Logica voor Alfa's en Informatici

Jan van Eijck, Centrum voor Wiskunde en Informatica (CWI), Amsterdam,
Elias Thijsse, Faculteit Humanistiek, Universiteit Tilburg
Gerard Vreeswijk, Departement Informatica, Universiteit Utrecht

November 2018

Inhoudsopgave

Voorwoord (1989)

Dit boek biedt een inleiding in de logica voor geïnteresseerden met een niet primair wiskundige achtergrond. Het boek is gebaseerd op cursusmateriaal dat in de loop van een aantal jaren ontwikkeld is in het werkverband Taal en Informatica van de Letterenfaculteit der Katholieke Universiteit Brabant in Tilburg. Daardoor is het speciaal geschikt voor mensen met belangstelling voor taaltheorie, informatica of een combinatie van die twee disciplines. Hoewel het boek bedoeld is als een 'inleiding voor iedereen' hebben wij gekozen voor een presentatie die de formele aspecten niet schuwt. Wij zijn van oordeel dat wie zijn aandacht beperkt tot filosofische achtergronden van de logica of tot toepassingen, slechts een aftreksel proeft van het vak. Een echte kennismaking kan de formele aanpak niet ontberen: logica zonder symbolen is geen logica.

Lezers die niet vertrouwd zijn met de wiskundige manier van denken verwachten dat het proza—ook het wetenschappelijk proza—dat zij tot zich nemen qua taalgebruik nauwelijks geheimen bevat. Bij boeken die geschreven zijn door auteurs met een β-inslag klopt dat uitgangspunt niet, en frustratie is vaak het gevolg. Stel even dat je iemand bent die in het verleden op deze manier gefrustreerd is geraakt. Dan volgt hier een simpel recept om dat soort frustratie in de toekomst te voorkomen. Ga eerst na of zeker boek wel voor je bedoeld is. Wanneer je nooit iets aan wiskunde of informatica heeft gedaan vallen er op die manier een heleboel boeken af. Jammer, maar wie graag in het diepe wil springen moet nu eenmaal eerst leren zwemmen. Om ervoor te zorgen dat dit boek je *niet* frustreert doe je er goed aan te beseffen dat kennismaken met formele methoden een vorm van mentale yoga is die vraagt om tijd, concentratie en toewijding. Je dient je de kost die je hier krijgt voorgeschoteld bladzijde voor bladzijde eigen te maken, met af en toe herkauwen voor de goede vertering. Wanneer je zich met die geesteshouding aan het werk zet zullen grote voldoening en blijvende vreugde uw deel zijn, niet in het minst omdat je zult merken dat allerlei literatuur op het gebied van logica, theoretische informatica en semantiek die eerst te hoog gegrepen was nu binnen uw bereik komt.

Als voorbereiding op wat je te wachten staat globaal iets over de inhoud.

– Hoofdstuk 1 geeft een algemene inleiding. Je leert één en ander over de achtergronden van vak, en maakt kennis met jargon en bewijstechnieken.

– Hoofdstukken 2 tot en met 4 geven een overzicht van wat je moet weten van verzamelingenleer om logica en berekenbaarheidstheorie te kunnen bedrijven. Hoofdstukken 2 en 3 geven de basisbegrippen en Hoofdstuk 4 bespreekt oneindigheid.

– Hoofdstuk 5 bespreekt welke problemen computers nog aankunnen en welke niet. Voor Hoofdstuk 5 is het al echt nodig dat je goed vertrouwd bent met het materiaal uit de voorgaande hoofdstukken!

– Hoofdstuk 6 legt uit wat logica is, en waar het in dat vak om gaat, en vervolgens presenteren de volgende hoofdstukken de twee standaard systemen die de basis vormen van de moderne logica: propositielogica (7, 8, 9) en predikatenlogica (10, 11, 12, 13, 14).

– Het laatste gedeelte is geheel gewijd aan toepassingen van de logica in de informatica. Er is een hoofdstuk over logisch programmeren en PROLOG, en er is een hoofdstuk over programmacorrectheid. In dit hoofdstuk leer je hoe logica kan worden gebruikt te bewijzen dat programma's voldoen aan een vooraf opgelegde specificatie.

Het boek is zowel geschikt voor zelfstudie als voor groepsonderwijs. Lesmateriaal voor een beginnerscursus kan gevormd worden uit hoofdstukken 1, 2, 4, 7, 8 en 9; voor een meer gevorderde cursus kan daarnaast ook geput worden uit Hoofdstuk 5 en een selectie uit de hoofdstukken over predikatenlogica. Er is gezorgd voor voldoende oefenstof in de vorm van opdrachten. De opdrachten variëren van gemakkelijk tot behoorlijk pittig. Als je de antwoorden bij *alle* opdrachten onmiddellijk 'ziet' bent je een uitzondering; in dat geval raden wij je aan onverwijld logica te gaan studeren. Maar je zult waarschijnlijk merken dat je sommige—of wellicht zelfs alle—opdrachten moeilijk vindt. In dat geval moet je bedenken dat er is geen koninklijke weg is tot de logica. Ook al leggen we in dit boek de rode loper voor je uit, het zou onsportief zijn om alle moeilijkheden die je onderweg kunt tegenkomen onder het kleed te vegen, en dat hebben wij dan ook niet gedaan.

Rest ons de aangename plicht om iedereen te bedanken die op enigerlei wijze aan de totstandkoming van dit boek heeft bijgedragen. Allereerst dank aan alle letteren-studenten in Tilburg op wie wij protoversies hebben kunnen uitproberen. Miriam Mulders, Martin Kammler en Emiel Krahmer zijn als student-assistenten nauw bij het ontstaan van dit boek betrokken geweest; wij danken hen voor hun bijdragen aan het antwoorden-gedeelte en voor hun stimulerend commentaar op de tekst. Dank ook aan onze Tilburgse collega's van het ITK voor hun commentaar op hoofdstuk 8 en hun hulp in meer algemene zin; we noemen met name René Ahn, Frens Dols, Hans-Peter Kolb, Erik-Jan van der Linden en Reinhard Muskens. Tenslotte wijzen wij er graag op dat Johan van Benthem, de persoon van wie wij allebei het vak hebben geleerd, indirect zijn onmiskenbare stempel heeft gedrukt op onze kijk op en presentatie van het materiaal in dit boek.

25 februari 1989

Jan van Eijck — Cambridge
Elias Thijsse — Loon op Zand

Editie 2011

Allereerst dank ik Jan van Eijck en Elias Thijsse voor het beschikbaar stellen van het manuscript van “logica voor alpha’s en informatici” ten behoeve van een in 2010 nieuw te ontwikkelen cursus “logica voor informatica”. Ook dank ik Wiebe van der Hoek, Jan-Willem Klop, John-Jules Meyer, Roel de Vrijer en Femke van Raamsdonk voor het beschikbaar stellen van materiaal. Zonder hen was deze heruitgave niet mogelijk geweest.

De oorspronkelijke tekst, “logica voor alpha’s en informatici,” werd in 1989 uitgegeven als cursusboek door Academic Service. Deze tekst is grotendeels ongewijzigd gelaten. Soms werd de tekst door mij aangevuld of geactualiseerd.

– Een sectie “bewijstechnieken” is ingevoegd omdat de ervaring leert dat 1e-jaars in het algemeen moeite hebben met het zelfstandig produceren van wiskundige bewijzen. Met name is vaak niet duidelijk tot o welk detail een bewijs dient te worden geleverd.

– Voor semantische tableaus is uitgelegd hoe deze ook, en misschien handiger, kunnen worden genoteerd in boomvorm.

– Logische bewijzen worden niet langer gegeven in Hilbert’s calculus maar via de voor veel studenten gemakkelijker en intuïtiever werkende Fitch diagrammen. De uitleg van Fitch diagrammen is grotendeels overgenomen uit [Hoek, van der 1995].

– De eindigheid- en substitutielemma’s zijn ingevoegd. Dit zijn cruciale lemma’s in predikatenlogica, omdat zij de spil vormen in gezond- en volledigheidsbewijzen.

– De sectie over beslisbaarheid is omgewerkt naar een aanmerkelijk groter hoofdstuk en hernoemd naar het fundamentelere begrip “berekenbaarheid”.

– Het deel over programma-correctheid is belangrijk uitgebreid met een methodiek om post-condities “omhoog” te werken in een programma, zodat er ook in de praktijk mee kan worden gerekend. Ook is programma-correctheid in verband gebracht met berekenbaarheid. Dit laatste is gedaan via de officiële weg, namelijk, door met behulp van Turing’s stop-functie aan te tonen dat er geen software kan bestaan die moderne programmatuur op correctheid controleert.

– Uiteraard is passief lezen alleen niet voldoende en zal moeten worden geoefend op het werkcollege en thuis. Daarom zijn er tal van extra oefeningen (met uitwerkingen) toegevoegd op plekken waar ik dat nodig en nuttig vond.

Ik hoop oprecht dat de oorspronkelijk auteurs, Jan van Eijck en Elias Thijsse niet al te zeer zijn gebruuskeerd door mijn goedbedoelde “verbeteringen”.

De titel van het dictaat

De titel van het oorspronkelijk boek (dat daarvoor, en nu weer, een dictaat is) heb ik ongewijzigd gelaten, niet alleen uit respect voor de auteurs, maar ook omdat ik het voor een gecombineerde opleiding van gametechnology en informatica zo’n toepasselijke titel vind.

Immers, de α-wetenschappen houden zich bezig met de producten van de menselijke geest. De α-wetenschappen zijn dus *geesteswetenschappen* (Eng.: philosophy). Daaronder vallen bijvoorbeeld linguïstiek, de afzonderlijke talen (zoals Nederlands, Engels, etc.), geschiedenis, wijsbegeerte, muziekwetenschappen, cultuurwetenschappen, kunstgeschiedenis, en creatieve wetenschappen, zoals industrieel ontwerp en, in het bijzonder, game-design. Gametechnology is dus deel een α-wetenschap.

Naast de α-wetenschappen kent men de β-wetenschappen en de γ-wetenschappen. De β-wetenschappen omvatten alle *natuurwetenschappen* (Eng.: science, of: natural sciences), zoals natuurkunde, scheikunde, biologie, aardwetenschappen en, in mindere mate, de wiskunde. (Immers, in de wiskunde kunnen ook kunstmatige constructies zoals bijvoorbeeld data-structuren worden bestudeerd, en valt daarmee voor een belangrijk deel weer onder de geesteswetenschappen.)

De γ-wetenschappen, tenslotte, behelzen mens- of gedragswetenschappen (Eng.: social sciences), zoals psychologie, sociologie, antropologie, economie, rechten, en communicatiewetenschappen.

Veel disciplines bevinden zich op een *grensvlak*. Denk bijvoorbeeld aan het interdisciplinaire gebied van de cognitiewetenschappen. Vraag jezelf eens af waar in de α-β-γ driehoek informatica zich zou moeten (of kunnen) bevinden.

1 november 2011

Gerard Vreeswijk Utrecht

Editie 2012

Wout Elsinghorst, Bart Jansen, Merel Rietbergen, en de studenten uit cursusjaar 2011-2012 wil ik hartelijk danken voor hun waardevolle commentaar op de 2011 editie. Op basis van jullie opmerkingen zijn waar mogelijk fouten gecorrigeerd, aanpassingen doorgevoerd en voorbeelden toegevoegd. Resterende fouten zijn uiteraard voor mijn rekening.

1 november 2012

Editie 2013

Fouten verbeterd, hier en daar sommen en/of uitwerken toegevoegd. Met dank aan studenten en student-assistenten uit het jaar 2012-2013. Het dictaat van 2012 kan worden gebruikt, mits je er alert op bent dat

hier en daar de nummering van opgaven e.d. kan zijn verschoven.

7 oktober 2013

Editie 2014

Modulo jaartallen dezelfde tekst als voor 2013.

1 september 2014

Hoofdstuk 1

Inleiding

1.1 Doelstelling

De logica heeft een respectabele traditie die teruggaat tot vóór Aristoteles. Tot aan het midden van de vorige eeuw was logica een integraal onderdeel van de filosofie, maar toen het vak door het werk van Gottlob Frege een nieuwe impuls kreeg zijn ook wiskundigen zich met de ontwikkeling ervan gaan bemoeien, met als gevolg dat logica het werktuig bij uitstek werd voor wiskundig grondslagenonderzoek. Toen in de jaren dertig van deze eeuw de informatica ontstond was het de wiskundige en logicus Alan Turing die de theoretische computerwetenschap stevig verankerde in de logica. Twintig jaar later werd de taalwetenschap op een nieuwe leest geschoeid door Noam Chomsky, die daarmee de basis legde voor een samenwerking tussen logici en taalkundigen. En aan het eind van de zestiger jaren liet de Amerikaanse logicus Richard Montague zien dat de betekenis van zinnen uit de taal van alledag in principe beschreven kan worden met machinerie uit de logica.

Deze snelle historische vogelvlucht maakt duidelijk dat logica vandaag de dag een vak is dat ligt op het grensgebied van verschillende wetenschapsterreinen, een vak ook waaraan door vogels van diverse pluimage wordt gewerkt. Gevolg is dat logica op tal van manieren kan worden gepresenteerd. Hoe verteerbaar het resulterende gerecht is hangt af van de interesse en de voorkennis van de consument. Dit dictaat beoogt een inleiding te geven in de logica voor lezers die niet bij uitstek wiskundig geschoold zijn, maar die belangstelling hebben voor informatica of taalkunde. Daarom is gekozen voor een benadering die de formele aspecten niet schuwt, maar die geen wiskundige voorkennis veronderstelt.

1.2 Gebruiken en noemen van uitdrukkingen

We beginnen dit inleidende hoofdstuk met een paar opmerkingen over het verschil tussen het *gebruiken* en het *noemen* van woorden. In de zin die we nu opschrijven wordt de letter *a* één keer genoemd en nergens gewoon gebruikt. Diezelfde letter wordt vier maal gebruikt maar nergens direct genoemd in deze zin. Het onderscheid tussen gebruiken en noemen (Eng.: use versus mention) is niet alleen van toepassing op schrijftekens, maar ook op woorden of taal-uitdrukkingen in het algemeen.

Wanneer we over eigenschappen van taal spreken moeten we vaak talige uitdrukkingen *noemen*. Voorbeeld:

> *In een Nederlandse zin kan* de *nooit het onderwerp zijn.*

Ga zelf na waarom in deze zin niet wordt gezondigd tegen het principe dat de zin uitdrukt.

Het *noemen* van taaluitingen gebeurt ook bij het gebruik van directe rede:

> *'Donder op, klootzak, uit m'n ogen', zei ze.* (1)

De aanhalingstekens worden gebruikt om een passage die letterlijk voorkwam te *noemen*. In de indirecte rede ziet het verslag er wellicht heel anders uit:

> *Ze zei dat het misschien verstandiger was als we elkaar voorlopig even niet zouden zien.*

Het onderscheid tussen noemen en gebruiken van een taal-uitdrukking zullen we in dit dictaat maken met behulp van cursivering (dat daarnaast gebruikt wordt voor nadruk), of door enkele of dubbele aanhalingstekens te gebruiken, of door een uiting uit de context van de lopende tekst te halen en er wit omheen te zetten (en eventueel een voorbeeldnummer ervoor).

Het onderscheid tussen noemen en gebruiken is context-afhankelijk. In de tekst wordt voorbeeldzin (1) genoemd in plaats van gebruikt. In de voorbeeldzin zelf wordt het gedeelte tussen enkele aanhalingstekens genoemd in plaats van gebruikt.

1.3 Formele en empirische wetenschap

Logica is een formele wetenschap. Dit wil zeggen: zintuiglijke kennisverwerving (en dus: kennis over de zintuiglijk waarneembare buitenwereld) speelt bij het verder ontwikkelen van deze wetenschap geen rol. De formele wetenschap bij uitstek is de wiskunde. Formele wetenschappen staan tegenover empirische wetenschappen. Een empirische wetenschap is een wetenschap waarbij empirische kennisverwerving wel een rol speelt. Voorbeelden van empirische wetenschappen zijn de natuurkunde, de biologie, de psychologie en de (empirische) taalkunde.

Voor het opstellen van een nieuwe logica-theorie is het onnodig om feitenmateriaal te verzamelen, enquêtes te houden, enzovoort. Net zo voor het opstellen van nieuwe theorieën in de wiskunde. Voor het *toepassen* van logische theorieën op de werkelijkheid, bij voorbeeld voor het ontwerpen van computerschakelingen, liggen de zaken anders. Datzelfde geldt voor het toepassen van wiskundige theorieën in de studie van de natuur. Zulke

toepassingen veronderstellen kennis van het aspect van de werkelijkheid waarop de formele theorie wordt toegepast.

Het onderscheid tussen formele en empirische wetenschap wordt wel uitgedrukt in termen van het filosofische begrippenpaar kennis *a priori* versus kennis *a posteriori*. Een bewering die *a priori* waar of onwaar is, is een bewering waarvan de waarheid in principe kan worden vastgesteld met behulp van methoden waarin zintuiglijke waarneming geen rol speelt. Een bewering die *a posteriori* waar of onwaar is, is een bewering waarvan de waarheid of onwaarheid alleen kan worden vastgesteld op grond van zintuiglijke ervaring. Logica, wiskunde en formele taalkunde houden zich bezig met *a priori* theorieën, terwijl vakken zoals empirische taalkunde, natuurkunde en biologie zich bezighouden met *a posteriori* theorieën.

Zoals de titel van dit dictaat aangeeft willen wij logica presenteren aan alfa's (in het bijzonder: mensen met belangstelling voor taalkunde, filosofie en creatieve wetenschappen) en aan informatici (in het bijzonder: mensen met belangstelling voor praktische informatica). Wat is het belang van het aanleren van formele vakken zoals logica (of formele taalkunde en automatentheorie) voor de beoefening van empirische taalkunde en voor de praktijk van de informatica?

Eerst het belang van logica voor de informatica. Het beschrijven van de *betekenis* van constructies uit programmeertalen met behulp van het begrippenapparaat uit de logica (of verruimingen daarvan) heeft vrucht gedragen. Het blijkt mogelijk om het *effect* van afzonderlijke programmeeropdrachten te beschrijven met behulp van *paren* van logische omschrijvingen. Hierbij geeft het tweede lid van een paar een omschrijving van de situatie die ontstaat wanneer de opdracht wordt uitgevoerd terwijl de computer in een toestand is waarin het eerste lid van het paar waar is. Door nu volgens de regels van deze omschrijvingsmethode, ontwikkeld door C.A.R. Hoare en R.W. Floyd, een programma te annoteren met logische formules kan een programmeur zich vergewissen van de correctheid van dat programma. In de tweede plaats is de logica inspiratiebron geweest voor een geheel nieuwe stijl van programmeren, het zogenaamde *declaratief programmeren* dat wordt beoefend in programmeertalen zoals PROLOG.

Dan het verband tussen logica en empirische taalkunde. Empirische taalkunde houdt zich bezig met de bestudering van in de loop van de geschiedenis langzaam geëvolueerde mensentalen zoals het Nederlands, het Russisch of het Engels. Logica en formele taalkunde houden zich bezig met *geconstrueerde* talen: de taal van de propositielogica, de taal van de predikatenlogica (twee talen waarmee je verderop zult kennismaken), programmeertalen zoals Java en C#, enzovoorts. Talen van het eerste soort zullen we vanaf nu *natuurlijke talen* noemen. Talen van het tweede soort heten *formele talen*.

Zo op het eerste gezicht lijkt het misschien alsof talen van het eerste soort weinig uitstaande hebben met talen van het tweede soort. Het zou kunnen zijn dat het Nederlands en de taal van de predikatenlogica evenveel of even weinig met elkaar gemeen hebben als een zwaluw en een Fokker F-27. Als dat het geval is, dan heeft het voor geïnteresseerden in de empirische taalwetenschap weinig zin om zich te verdiepen in formele vakken. Aan de andere kant: als het mogelijk is om een theorie te ontwerpen die de overeenkomsten tussen zwaluwen en Fokker Friendships blootlegt, dan moet dat wel een erg fundamentele theorie zijn, juist *omdat* deze twee soorten van dingen op het eerste gezicht weinig gemeen lijken te hebben. Net zo: een theorie die iets interessants weet te zeggen met betrekking tot zowel natuurlijke talen en formele talen kan zeer fundamentele inzichten verschaffen.

Logica is op minstens drie punten relevant voor de empirische taalkunde. In de eerste plaats heeft de logica *formele methoden* ontwikkeld om te definiëren wat de uitdrukkingen zijn van een bepaalde *symbooltaal*. Deze definitie-methoden—die ook gebruikt worden om programmeertalen te definiëren—zijn van groot nut gebleken voor het bestuderen van de uitdrukkingen van een natuurlijke taal (het Nederlands, het Engels); niet dat je de uitdrukkingen van het Nederlands kunt 'afbakenen' zoals je de formules van een logische taal of de klasse van syntactisch correcte Java-programma's kunt afbakenen, maar je hebt dezelfde definitie-methoden nodig (al heb je er niet genoeg aan).

In de tweede plaats zijn logische talen vanwege hun grote precisie zeer geschikt om over betekenis-aspecten van natuurlijke taal te spreken. De uitdrukkingen van de symbooltalen die in de logica zijn ontwikkeld zijn altijd maar voor één manier van lezen vatbaar. Daar zijn die uitdrukkingen op gebouwd, zogezegd. Dat dit laatste bij natuurlijke taal niet het geval is moge blijken uit de volgende voorbeelden:

Lolita was jong en mooi of verdorven. (2)

Iedereen gelooft mij niet. (3)

Logische talen kunnen hier gebruikt worden als medium voor precisering van dubbelzinnige uitdrukkingen uit de natuurlijke taal.

Van voorbeeldzin (2) zijn bij voorbeeld de volgende twee preciseringen mogelijk in de taal van de propositielogica:

- $(p \wedge (q \vee r))$
- $((p \wedge q) \vee r)$

Hierbij staan de letters p, q en r respectievelijk voor 'Lolita was mooi', 'Lolita was jong' en 'Lolita was verdorven'; $\wedge$ staat voor 'en' en $\vee$ voor 'of'.

Voor voorbeeldzin (3) schaft de predikatenlogica raad. Hier zijn de twee mogelijke parafrases:

- $\forall x \neg Gxm$
- $\neg \forall x Gxm$

Legenda: $\forall x \ldots$: "voor alle x geldt dat…"; $\neg$: "niet"; Gxm: "x gelooft mij" (dat wil zeggen: "x gelooft wat ik zeg"). Nadere details volgen in de hoofdstukken 5 en 6.

De mogelijkheden die de logica biedt om over betekenisaspecten van taal te spreken gaan overigens nog veel verder, en daarmee komen we bij het derde raakvlak tussen logica en empirische taalkunde. In de logica is een zeer precieze uitleg voorhanden van het begrip 'betekenis' voor de formules uit een logische taal. Het ligt voor de hand om te proberen die uitleg (of een toepasselijke verruiming ervan) over te planten naar natuurlijke taal. Onderzoek in dit kader—onderzoek naar de "modeltheoretische semantiek van natuurlijke taal"—is zeer vruchtbaar gebleken, en dit onderzoek wordt momenteel toegepast in programma's die computergebruikers de mogelijkheid moeten bieden om in gewoon Engels of Nederlands te communiceren met een computer.

Wanneer we even afzien van aspecten van woordvorming of akoestische representatie (dat wil zeggen, wanneer we fonetiek, fonologie en morfologie buiten beschouwing laten), dan vallen er in de studie van natuurlijke talen de volgende aspecten te onderscheiden:

- **syntaxis:** de studie van zinnen, zinsbouw, zinsontleding, grammaticaliteit;
- **semantiek:** de studie van uitdrukkingen van de taal in relatie tot wat ze uitdrukken (hun betekenis);
- **pragmatiek:** de studie van uitdrukkingen, inclusief hun betekenis, in diverse gebruikscontexten.

De syntaxis, semantiek en pragmatiek van een taal verhouden zich zoals aangegeven in Figuur 1.1. Dit plaatje maakt duidelijk wat de *hiërarchie* is tussen de aspecten 'syntaxis', 'semantiek' en 'pragmatiek'. Het bestuderen van de semantiek van een taal veronderstelt een grote mate van inzicht in de syntaxis, maar andersom is dat veel minder het geval. Het bestuderen van de pragmatiek van een taal veronderstelt inzicht in syntaxis en semantiek van die taal, en van de samenhang daartussen, maar niet of nauwelijks andersom. Het plaatje suggereert misschien ook een hiërarchie in respectabiliteit van de drie disciplines 'syntaxis', 'semantiek' en 'pragmatiek'. Op grond van het plaatje zou je syntaxis kunnen beschouwen als hulpje van de semantiek, en semantiek + syntaxis als hulpje bij de pragmatiek. In de praktijk van het onderzoek van natuurlijke talen liggen de zaken echter heel anders: syntaxis is relatief het verst ontwikkelde vak, semantiek is een goede tweede, maar helaas: de pragmatiek is een zorgenkindje dat nog niet zo goed wil groeien.

Dat het heel goed mogelijk is syntaxis te bedrijven zonder je om semantiek te bekommeren wordt duidelijk in de formele taaltheorie. Hier wordt een (formele) *taal* doorgaans beschouwd als een verzameling uitdrukkingen, zonder dat de leden van die verzameling een of andere plausibele betekenis hoeven te hebben.

Formele talen waarbij het wel zin heeft om te spreken over de betekenis van de uitdrukkingen van de taal zijn er ook. Voorbeelden zijn: de taal van de propositielogica, die van de predikatenlogica, en de programmeertaal C#. Ook bij natuurlijke talen gaan we ervan uit dat welgevormde uitdrukkingen van zo'n taal in principe dragers van betekenis zijn. De in de syntaxis van zo'n taal bestudeerde taaluitingen hebben een *betekenis* die in principe achterhaald kan worden door te kijken naar de betekenis van de woorden die erin voorkomen. Sommige woorden in een zin verwijzen naar *objecten* in de wereld: "Prins Claus", "de vliegbasis Gilze-Rijen". Andere verwijzen naar *eigenschappen van objecten*: "rood", "ziek". Weer andere slaan op *relaties tussen objecten*: "bewondert", "bemint", "haat". Nog weer andere hebben de logische functie van 'bindmiddel': "niet", "en", "of", "elke".

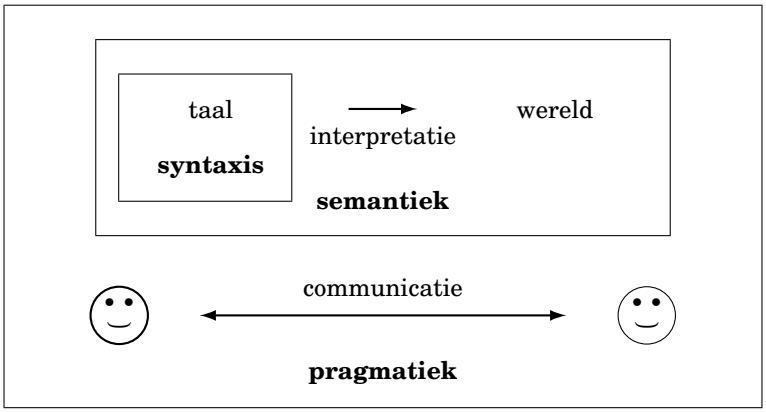
Door de informatie die de verschillende onderdelen van een zin leveren te combineren krijgen we de betekenis van de zin als geheel. We begrijpen op deze manier "Jan haat de vliegbasis Gilze-Rijen" op grond van het feit dat we weten waar "Jan" en "de vliegbasis Gilze-Rijen" op slaan, en welke relatie met "haat" wordt aangeduid. *Begrijpen* van de zin is overigens nog iets anders dan weten of de zin waar is. Begrijpen van een zin is veeleer iets in de trant van: weten *onder welke omstandigheden een zin waar is*, en dat is iets heel anders. We kunnen immers weten wat "haten" betekent zonder dat we precies weten wie wie of wat haat. Het bedrijven van semantiek is een vorm van begripsanalyse, en heeft niets te maken met verificatie van feiten of verwerven van kennis over de werkelijkheid.

Voor de conceptuele analyse die door semantici wordt bedreven is het meestal niet nodig om de werkelijkheid als geheel in de beschouwingen te betrekken. Vaak is alleen een klein stukje daarvan relevant voor het begrijpen van een tekst, en in een rationele reconstructie kunnen we dat gedeelte schematiseren tot een zogenaamd *model*. We zullen zien dat in de formele logica de begrippen "syntaxis", "semantiek", "interpretatie" en "model" exact gedefinieerd zijn.

1.4 Jargon

In dit boek zullen we—terwille van de oriëntatie-mogelijkheden in de veelal Engelstalige vakliteratuur—naast het Nederlandse jargon ook waar nodig wat Engelse termen laten vallen, zonder ons al te zeer te bezondigen aan het oplepelen van half-verteerde brokstukken Engels vocabularium met een Nederlands syntactisch sausje, niet ongebruikelijk onder Nederlandse logici, taalkundigen en (vooral) informatici. In plaats van "deze propositie is vals" zullen we dus zeggen: "deze bewering is onwaar", en "een value-assignment voor de variabelen" zullen we vervangen door: "het toekennen van waarden aan de variabelen", of "een bedeling voor de variabelen". Met het uitleggen van (nuttig, want het alledaags spraakgebruik aanvullend en preciserend) jargon maken we in deze paragraaf alvast een begin.

Vele taalfilosofen maken onderscheid tussen *zinnen* (Eng.: sentences), *uitspraken* of *beweringen* (Eng.: statements), *proposities* (Eng.: propositions), *standen van*



Figuur 1.1: Syntaxis, semantiek en pragmatiek van een taal.

zaken (Eng.: states of affairs), en nog zo het één en ander. Soms is het gemakkelijk om te kunnen zeggen dat verschillende *zinnen* een en dezelfde *bewering* kunnen uitdrukken. Bij voorbeeld: "het sneeuwt" en "il neige" drukken hetzelfde (dezelfde bewering) uit. Over de andere onderscheidingen zullen we ons niet druk maken.

Vaak wordt onderscheiden tussen de *extensie* en de *intensie* van een woord of zin. Deze termen stammen uit de traditionele filosofie. De *intensie* van een woord is—ongeveer—de gedachte-inhoud van dat woord, ofwel: het geheel van wat een gebruiker van dat woord moet weten om het woord correct te kunnen gebruiken. De intensie of gedachte-inhoud van *de koning(in) der Nederlanden* is zoiets als "de persoon die in Nederland de erfelijke functie van staatshoofd bekleedt".

De *extensie* van een woord is datgene waar dat woord naar verwijst. De extensie van *de koningin der Nederlanden* is Beatrix. De extensie van *Nederlandse hoogleraar* is de verzameling van alle individuen die nu in Nederland de functie van hoogleraar bekleden. Soms worden in plaats van *extensie* ook de termen *referentie*, *verwijzing* of *denotatie* gebruikt. We zeggen ook: de spreker *refereert*/*verwijst* naar een (buitentalig) object met behulp van een (talige) uitdrukking. Het object waarnaar met een uitdrukking wordt verwezen wordt de *referent* van die uitdrukking genoemd.

De extensie van *eenhoorn* bevat niets. In het wiskundige jargon dat je spoedig (weer) vertrouwd in de oren zal klinken: het is de lege verzameling. De reden is dat er geen eenhoorns zijn (hoewel ze voortdurend opdraven om als voorbeelden te dienen in taalfilosofische discussies over intensies en extensies).

In de moderne semantiek wordt er een truc toegepast om *intensie* te definiëren in termen van *extensie*. De intensie van *de koningin der Nederlanden* is nu een recept om per situatie de juiste extensie uit te kiezen. Dus, variërend in tijd tussen 1945 en nu: Wilhelmina, Juliana, Beatrix. De intensie van *eenhoorn* kan in situaties die voldoende van onze prozaïsche werkelijkheid verschillen wel degelijk een verzameling zachtaardige, schuwe en ge-eenhoornde viervoeters opleveren. Vandaar dat we kunnen zeggen dat de betekenis (de intensie) van *gevleugeld paard* en *eenhoorn* van elkaar verschillen, terwijl de verwijzing-hier-en-nu (de extensie) in beide gevallen hetzelfde is, namelijk de lege verzameling.

Een terminologisch onderscheid dat van Frege stamt is dat tussen *Sinn* (Eng.: sense) en *Bedeutung* (Eng.: reference). Hierover zijn bibliotheken volgeschreven (zie bij voorbeeld [Dummett 1973] voor een uitvoerige beschouwing en verdere literatuurverwijzingen). In de moderne semantiek gaat men er meestal van uit dat het nieuwerwetse onderscheid tussen *intensie* (in de zin van: recept) en *extensie* Frege's onderscheid op een acceptabele manier preciseert. Zie voor deze en verwante begripsonderscheidingen ook [Lewis 1972].

Een begrip, tenslotte, waarover in kringen van empirisch taalkundigen veel gesproken wordt is het begrip *logische vorm*. In de Transformationeel-Generatieve traditie in de taalwetenschap (dat wil zeggen, de Chomsky-traditie) slaat dit op een syntactische representatie van een zin, met daarin extra informatie gecodeerd die de dubbelzinnigheden eruit haalt. Daarbij kan het bij voorbeeld gaan om de dubbelzinnigheden die veroorzaakt worden door de aanwezigheid van pronomina of van *logische operatoren* (die we hierboven 'logisch bindmiddel' genoemd hebben). Voorbeelden:

Iedereen zei dat hij *kwam.*

Alle hout is geen timmerhout.

Zoals we hierboven al hebben aangegeven kan een dergelijke desambiguering ook bewerkstelligd worden door vertaling in een geschikte logische taal. Dergelijke desambiguerende vertalingen ('analyses') worden daarom ook wel "logische vormen" genoemd. Echter: logische vormen zijn nu niet meer *uniek*. Ze hangen nu in de eerste plaats van het gekozen analyse-medium af (propositielogica, predikatenlogica, of nog een andere logische taal). In de tweede plaats zijn er zelfs binnen een formele taal in het algemeen meerdere logische formules mogelijk, omdat die formules *logisch gelijkwaardig* (of: *logisch equivalent*) zijn. Twee formules zijn logisch

equivalent wanneer die formules dezelfde interpretatie krijgen, ongeacht hoe het model eruit ziet waarin wordt geïnterpreteerd. De volgende twee *logische vormen* (vertalingen in de predikatenlogica) voor "Niemand houdt van Marietje" zijn bij voorbeeld equivalent:

– $\neg\exists x\, Hxm$ (letterlijk: "niet: iemand houdt van Marietje")

– $\forall x \neg Hxm$ (letterlijk: "Iedereen houdt-niet-van Marietje")

In discussies over 'logische vorm' zijn spraakverwarringen aan de orde van de dag omdat de discussiepartners vaak niet hetzelfde bedoelen met de term. Een grondige vertrouwdheid met logica kan je tegen veel vruchteloos gekissebis behoeden. Voor we aan de logica zelf toe zijn moet er echter wat wiskundig voorwerk worden verricht.

1.5 Getalverzamelingen

In de wiskunde en informatica worden dubbele letters voor veelvoorkomende getallenverzamelingen gebruikt:

- $\mathbb{N}$: de verzameling van **natuurlijke getallen**: alle positieve gehele getallen dus $1, 2, 3, \ldots$.

  Waarschuwing: in de berekenbaarheidstheorie (blz. 59) wordt 0 ook als natuurlijk getal beschouwd. Ook bij programmeren wordt er geteld vanaf 0. (Dit heeft te maken met hoe programmeertalen als Java en C hun array-elementen indiceren.) In alle andere takken van de wiskunde en de informatica begint men te tellen vanaf 1.

- $\mathbb{Z}$: de verzameling van **gehele getallen** $\ldots, -2, -1, 0, 1, 2, \ldots$

- $\mathbb{Q}$: de verzameling van **rationele getallen**: alle breuken, dat wil zeggen getallen die geschreven kunnen worden als $\frac{p}{q}$, waar p en q gehele getallen zijn en $q \neq 0$.

- $\mathbb{R}$: de verzameling van **reële getallen**: alle getallen, ook getallen die geen breuken zijn (zoals $\sqrt{2}$ en π).

Opmerkingen:

1. Ga na dat elke volgende verzameling meer bevat dan de vorige verzameling. Dus $\mathbb{N}$ is een deelverzameling van $\mathbb{Z}$, en de laatste is weer een deelverzameling van $\mathbb{Q}$, enzovoort.

2. De $\mathbb{Z}$ komt van het Duitse woord "Zahlen". De $\mathbb{Q}$ komt van het alternatieve woord voor breuk: "quotient".

3. Getallen die niet in $\mathbb{Q}$ zitten heten *irrationaal* ("niet gebroken").

4. Er bestaan meer getallenverzamelingen waar dubbele letters voor worden gebruikt: $\mathbb{A}$ voor de verzameling van algebraïsche getallen (zit tussen $\mathbb{Q}$ en $\mathbb{R}$ in), $\mathbb{C}$ voor de verzameling van complexe getallen (bevat meer dan $\mathbb{R}$). We besteden hier verder geen aandacht aan.

In de volgende opgaven kun je een beetje oefenen met de verschillende getalverzamelingen.

Opgave 1.1 Waar of onwaar?

1. $0 \in \mathbb{N}$.	6. $1/2 \in \mathbb{Z}$.	11. $\sqrt{2} \in \mathbb{Q}$.	16. $\pi \in \mathbb{R}$.
2. $1 \in \mathbb{N}$.	7. $4 \in \mathbb{Z}$.	12. $0.1 \in \mathbb{Q}$.	17. $-5 \in \mathbb{R}$.
3. $-1 \in \mathbb{N}$.	8. $\pi \in \mathbb{Z}$.	13. $\pi \in \mathbb{Q}$.	18. $0 \in \mathbb{R}$.
4. $2 \in \mathbb{N}$.	9. $-4 \in \mathbb{Z}$.	14. $0 \in \mathbb{Q}$.	19. $\sqrt{7} \in \mathbb{R}$.
5. $1/2 \in \mathbb{N}$.	10. $0.1 \in \mathbb{Z}$.	15. $e \in \mathbb{Q}$.	20. $\sqrt{7}/\pi \in \mathbb{R}$.

Noot: het getal e is het grondtal van de natuurlijke logaritme en is ongeveer gelijk aan 2.71828. Het getal e is irrationaal.

Opgave 1.2 Het is verleidelijk getalverzamelingen te schrijven (of zelfs te definiëren) als

$$\begin{aligned} \mathbb{N} &=_{Def} \{1, 2, 3, \ldots\}, \\ \mathbb{Z} &=_{Def} \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}, \\ &\ldots \end{aligned}$$

Waarom is dit uiteindelijk toch geen goed idee? (Einde opgave.)

Verder kom je ook nog wel eens het volgende tegen:

– $\mathbb{N}_0$: de verzameling natuurlijke getallen, uitgebreid met 0.

– $\mathbb{Q}^+$: de verzameling positieve breuken.

– $\mathbb{Q}_0^+$: de verzameling positieve breuken, inclusief 0.

– $\mathbb{R}^+$: de verzameling positieve reële getallen.

– $\mathbb{R}_0^+$: de verzameling positieve reële getallen, inclusief 0.

Als je regelmaat in de notatie ziet wordt deze makkelijker te onthouden.

Opgave 1.3 Soms komt men het symbool $\mathbb{Z}^+$ tegen. Bediscussieer het nut hiervan.

1.6 Bewijstechnieken

In de wiskunde en informatica word je regelmatig gevraagd iets te bewijzen. Dat "iets" is dan bijvoorbeeld een stelling, een lemma, een propositie, een bewering die gedaan wordt in een opgave, of gewoon een los resultaat. De vraag wordt vaak gesteld in de vorm "bewijs dat ...," of "toon aan dat ...," of "laat zien dat ...".

In het begin kunnen dergelijk vragen intimiderend zijn of stress oproepen. Je weet dan namelijk nog niet goed wat er van je verwacht wordt, met name weet je vaak niet hoe gedetailleerd of rigoureus een bewijs er uit moet zien. Deze sectie behandelt een aantal bewijstechnieken.

We zullen behandelen: bewijs door gevalsonderscheid, bewijs door contrapositie, bewijs uit het ongerijmde, bewijs door volledige inductie, niet-constructieve bewijzen, en uniciteitsbewijzen. Alvorens deze technieken te gaan bespreken is het misschien goed eerst wat aandacht te geven aan de volgende vraag.

Wat is een bewijs?

Er zijn verschillende manieren om te vertellen waar een bewijs aan zou moeten voldoen. We zouden bijvoorbeeld een lang en uitputtend betoog kunnen gaan opzetten over regels die je zou moeten volgen en principes waar je aan zou moeten houden, om een goed bewijs te kunnen leveren. Dat is de normatieve aanpak.

Het probleem met een normatieve aanpak is dat deze zou moeten werken in verschillende situaties (thuis, voor het bord, tijdens een tentamen). Maar dat gaat niet. Het is nu eenmaal zo dat in verschillende situaties verschillende typen bewijzen worden gevraagd. Zo zal een bewijs opgeschreven in een cursusboek er anders uit (moeten) zien dan een mondeling bewijs dat gegeven wordt met behulp van spraak, krijt, een schoolbord, en misschien nog wat lichaamstaal (armgebaren, dingen aanwijzen).

Ook worden bewijzen op verschillende niveau's gegeven. Zo verloopt een formeel bewijs in, bijvoorbeeld, de object-taal van de natuurlijke deductie, op een volledig ander niveau dan een informeel en schetsmatig bewijs in, bijvoorbeeld, de berekenbaarheidstheorie.[1]

Uiteindelijk is een bewijs een sociaal iets. Dus uiteindelijk komt de vraag neer op: wanneer accepteert iemand, of een groep, iets als een bewijs? Dat hangt af van de ontvanger. Als een bewijs bijvoorbeeld "live" wordt gegeven, hebben toehoorders een kans om in te breken en aanvullende uitleg te vragen, net zo lang totdat iedereen overtuigd is, of totdat het "bewijs" ter plekke niet meer gerepareerd kan worden en er geen bewijs blijkt te zijn. Ook kan het zijn dat één van beide partijen moe wordt, en het opgeeft. In dat laatste geval is er natuurlijk ook geen bewijs. Als het podium een wetenschappelijk tijdschrift is, zal een bewijs veel preciezer moeten zijn omdat de schrijver zal moeten anticiperen op eventuele vragen van de lezer die, op het moment dat deze het bewijs tot zich neemt, niet meer de kans heeft om aanvullende uitleg te vragen. Maar ook bij gepubliceerde bewijzen komt het voor dat pas jaren nadien een fout wordt ontdekt. Kurt Gödel, zonder twijfel de belangrijkste logicus uit de 20e eeuw, bewees in 1932 bijvoorbeeld dat de geldigheid van een bepaalde klasse van volzinnen in de Peano rekenkunde, bekend als $\langle \exists^* \forall^2 \exists^*, all, (0) \rangle$, beslisbaar was.[2] In de laatste zin van zijn artikel beweerde Gödel dat het bewijs van die bewering ook opging voor een grotere klasse, $\langle \exists^* \forall^2 \exists^*, all, (0) \rangle_=$, die ook nog formules met gelijkheden bevat. Echter, midden jaren '60 toonde de Noorse wiskundige Stål Aanderaa aan dat Gödel's bewijs niet opging voor die grotere klasse. In 1982 toonde de Amerikaanse logicus en filosoof Warren Goldfarb aan dat deze grotere klasse inderdaad onbeslisbaar was.[3]

Was Kurt Gödel dus een rommelaar? Nee, natuurlijk niet. Het was een top-logicus die tijdens zijn werk ook wel eens een foutje maakte. Waar gehakt wordt vallen spaanders. Onthoud maar dat de notie "bewijs" een relatief begrip is, en dat zelfs de beste bewijzen niet per definitie onweerlegbaar zijn. Misschien haalt dat een beetje van de stress af.

We laten wat bewijstechnieken de revue passeren, te beginnen met het bewijs door gevalsonderscheid.

Bewijs door gevalsonderscheid

We beginnen met *bewijs door gevalsonderscheid* omdat het goed laat zien hoe het geven van een bewijs werkt. Het volgende voorbeeld is niet spannend, maar wel erg leerzaam.

Voorbeeld 1.1 Zoals je weet wordt de absolute waarde van $x \in \mathbb{R}$ meestal gedefinieerd als

$$|x| =_{Def} \begin{cases} x & x \geq 0, \\ -x & \text{anders.} \end{cases}$$

Er zijn kleine variaties mogelijk, bijvoorbeeld $|x| = x$ als $x > 0$ en $|x| = -x$ als $x \leq 0$. Laten we toch maar de eerste definitie gebruiken. Opdracht: Bewijs dat $|x| \geq x$.

Uitwerking: omdat de notie absolute waarde gedefinieerd werd met gevalsonderscheid lig het voor de hand het bewijs ook met gevalsonderscheid te geven. Zij $x \in \mathbb{R}$ willekeurig. Er zijn twee mogelijkheden: $x \geq 0$ of $x < 0$. Deze mogelijkheden zijn uitputtend. (I.e., andere mogelijkheden zijn er niet.)

1. Als $x \geq 0$, dan per definitie $|x| = x$. Omdat zeker geldt $x \geq x$, is voor dit geval het gevraagde bewezen.

2. Als $x < 0$, dan per definitie $|x| = -x$. We moeten dus controleren of $-x \geq x$. Dat is het geval, want $-x$ is positief en x is negatief.

Het gevraagde is nu bewezen voor alle mogelijke waarden $x \in \mathbb{R}$. □

Opgave 1.4 Bewijs dat voor alle $x \in \mathbb{R}$ geldt dat $|x| \geq 0$.

Voorbeeld 1.2 Bewijs: $|x + y| \leq |x| + |y|$.

Als we het bewijs met gevalsonderscheid gaan geven zijn er acht gevallen te onderscheiden, afhankelijk van $x \geq 0$, $y \geq 0$ en $x + y \geq 0$, die onafhankelijk kunnen voorkomen. Als een expressie $z \geq 0$ noteren we dat met "+". (We zouden dit preciezer met "$+_0$" moeten noteren, wat we niet doen.) Als een expressie $z < 0$ noteren we dat met "−". We krijgen dan de volgende acht gevallen.

[1] Op Wikipedia is er een aardige pagina "list of long proofs" over de langste bewijzen ooit gegeven. Volgens die pagina wordt sinds 2000 een bewijs als ongebruikelijk lang gezien als het meer dan 500 kantjes beslaat. De pagina vermeldt niet wanneer een bewijs wordt geclassificeerd als "gewoon" lang.

[2] De notie beslisbaarheid komt aan de orde in het hoofdstuk over berekenbaarheid.

[3] Wikipedia: "list of incomplete proofs".

x	y	$x+y$	$\lvert x\rvert$	$\lvert y\rvert$	$\lvert x+y\rvert$	
$+$	$+$	$+$	x	y	$x+y$	$\surd$
$+$	$+$	$-$	kan niet voorkomen			
$+$	$-$	$+$	x	$-y$	$x+y$	$\surd$ (*)
$-$	$+$	$+$	$-x$	y	$x+y$	$\surd$ (**)
$+$	$-$	$-$	x	$-y$	$-x-y$	$\surd$
$-$	$+$	$-$	$-x$	y	$-x-y$	$\surd$
$-$	$-$	$+$	kan niet voorkomen			
$-$	$-$	$-$	$-x$	$-y$	$-x-y$	$\surd$

We bewijzen één geval: (*). Het bewijs van de andere gevallen verloopt analoog.

In geval (*) moeten we laten zien dat $x+(-y)\geq x+y$. Dit komt neer op het aantonen van $0\geq 2y$ wat waar is, want $y<0$. Dit was het bewijs voor het geval (*). Zoals gezegd verloopt het bewijs van de andere gevallen analoog.

Als je iets bewijst, en er komen veel dezelfde deel-gevallen voor waarvan de bewijzen ook nog eens op elkaar lijken, mag het gehele bewijs dan afgesloten worden met een opmerking dat het bewijs van de resterende gevallen analoog verloopt, zonder die bewijzen ook daadwerkelijk te geven? Ja, dat mag. Van de lezer mag worden verwacht dat, als de resterende bewijzen werkelijk analoog verlopen, hij (of zij) die bewijzen juist daarom ook zelf kan reconstrueren.

Opgave 1.5 Bewijs geval (**).

Hoewel $|x+y|\leq|x|+|y|$ een uitspraak over absolute waarden is, kon in dit speciale geval het gevraagde toch zonder gevalsonderscheid worden bewezen:

$$\begin{aligned}(|x+y|)^2 &= (x+y)^2\\ &= x^2+2xy+y^2\\ &\leq x^2+|2xy|+y^2\\ &= |x|^2+2|x||y|+|y|^2\\ &= (|x|+|y|)^2.\end{aligned}$$

Omdat voor alle $u,v\geq 0$ geldt:

$$u^2\leq v^2\Rightarrow u\leq v, \tag{4}$$

geldt nu ook $|x+y|\leq|x|+|y|$. Het bewijs van (4) laten we ter oefening aan de lezer (Opgave 1.6).

Opgave 1.6 1. Bewijs

$$-v\leq u\leq v \text{ als en alleen als } |u|\leq v.$$

2. Gebruik het vorige onderdeel om een alternatief bewijs te geven van $|x+y|\leq|x|+|y|$.

Ledigerwijs

Deze sectie behandeld geen bewijstechniek, maar meer een verschijnsel dat vaak als randgeval opduikt in bewijsconstructies, namelijk, "lege waarheden". Een "lege waarheid" (Eng.: "vacuous truth") beweert iets over niet bestaande elementen. Zo'n bewering is altijd waar. In de Nederlandse taal bestaat het woord "ledigerwijs" officieel niet. Wij gebruiken het hier als een gekunstelde vertaling van het Engelse begrip "vacuously".

Voorbeelden van beweringen die ledigerwijs gelden zijn de volgende:

1. "Ik heb al mijn groenten opgegeten." is een lege ware bewering als je geen groenten opgeschept kreeg.
2. "Elke keer als ik piano speelde in het concertgebouw schreven de kranten zeer lovend over mij." is een lege ware bewering als je nooit piano speelde in het Amsterdamse concertgebouw.
3. "Elke keer als jij wint, krijg jij je inzet verdubbeld terug." is een lege ware bewering bij balletje-balletje. (Want bij balletje-balletje zorgt men wel dat jij niet wint.)

Enzovoort.

Opgave 1.7 Welke van de volgende beweringen zijn ledigerwijs waar? (Niet eerder bij de antwoorden kijken voordat alles is beantwoord.)

1. Alle nulpunten van de parabool $y=x^2+4$ zijn positief.
2. Alle nulpunten van de parabool $y=x^2+4$ zijn negatief.
3. Alle elementen van de verzameling $A=\{x\in\mathbb{R}\mid x^2=2\}$ zijn positief.
4. Alle elementen van de verzameling $A=\{x\in\mathbb{Q}\mid x^2=2\}$ zijn positief.
5. Alle geheeltallige oplossingen (x,y,z) van $x^2+y^2=z^2$ voldoen aan $x+y\leq z+2$.
6. Alle niet-triviale geheeltallige oplossingen van $x^3+y^3=z^3$ voldoen aan $x+y\leq z+3$.

   Hint: volgens de laatste stelling van Fermat die rond 1995 bewezen werd door Andrew Wiles en anderen, bezit $x^n+y^n=z^n$ geen niet-triviale geheeltallige oplossingen zodra $n>2$.[4]
7. Alle geheeltallige oplossingen van $x^3+y^3=z^3$ voldoen aan $x+y\leq z+3$.

Bewijs met een invariant

Beschouw de volgende processen: het betegelen een keukenvloer; het trekken van ballen uit een vaas; het spelen van boter, kaas en eieren. Voorbeelden van invarianten kunnen zijn: het aantal gelegde tegels + het aantal open plekken; de pariteit ("evenheid") van het aantal overgebleven witte ballen; het aantal gedane zetten + het aantal open velden.

Definitie 1.1 (Invariant) *Een* invariant *is een eigenschap die blijft gelden tijdens het hele proces.*

[4] Oplossingen als $(x,y,z)=(0,0,0)$ en $(x,y,z)=(0,1,1)$ en $(x,y,z)=(-1,1,0)$ worden triviaal genoemd.

Een invariant kan vaak helpen bij het doen van uitspraken over eigenschappen in dat proces. Invarianten worden bijvoorbeeld intensief gebruikt in de theorie van programma-correctheid. In het bijzonder worden daar naar zogenaamde *lus-invarianten* gezocht om programma's correct te bewijzen. (Zie verder Sectie 17.3 op blz. 204.)

Voorbeeld 1.3 Er is een onbetegelde vierkante keukenvloer met een zijde van n (kleine) vierkanten. De bedoeling is deze vloer te betegelen met domino's van 2×1. In de linker-bovenhoek en de rechter-onderhoek liggen twee vierkanten die niet betegeld kunnen worden in verband met de aanwezigheid van uitbouw en/of verwarmingsbuizen. Bewijs dat voor geen enkele n de keukenvloer met domino's betegeld kan worden.

Oplossing: maken een gevalsonderscheid. Als n oneven is dan is n^2 oneven, en $n^2 - 2$ dus ook. Een vloer met een oneven aantal vierkanten is nooit te betegelen met tegels die elk 2 vierkanten groot zijn.

Als n even is dan kleuren we de vloer in gedachten als een schaakbord: om en om wit en zwart, met de rechter onderhoek wit. De twee verwarmingsbuizen staan in een wit veld. Er moeten dus twee zwarte velden méér betegeld worden dan witte velden. Omdat elke tegel één zwart en één wit veld bedekt, zullen we, hoe we ook tegelen, altijd twee zwarte velden overhouden. Een betegeling is dus niet mogelijk. We hebben gebruik gemaakt van de invariant

> aantal onbetegelde zwarte velden = aantal onbetegelde witte velden + 2

Deze uitspraak is in het begin waar, en blijft gelden bij het leggen van een tegel. De uitspraak geldt dus voor alle gedeeltelijke betegelingen, en er is geen betegeling waarin geen onbetegelde zwarte velden over zijn.

Opgave 1.8 (De draak) Een draak heeft 100 koppen. Een ridder kan in één houw tegelijkertijd 15, 17, 20, of 5 koppen afhakken. In elk van deze gevallen groeien respectievelijk 24, 2, 14, en 17 koppen aan. De draak sterft als alle koppen er af zijn. Bewijs dat de draak op deze manier nooit kan sterven. (Hint: bekijk het aantal koppen dat er na elke slag bijkomt of afgaat. Wat hebben deze aantallen gemeen? Heeft 100 deze eigenschap ook?) Benoem de invariant.

Opgave 1.9 (De koffiekan) In een koffiekan zitten 25 zwarte en 25 witte bonen. De volgende handeling wordt uitgevoerd zolang dat mogelijk is: haal twee bonen uit de kan; als ze dezelfde kleur hebben, stop dan een zwarte boon in de kan; als ze verschillend van kleur zijn, stop dan de witte terug in de kan. Er is een voldoende voorraad extra zwarte bonen. Vraag: wat is de kleur van de laatste boon in de kan? Hints:

1. Beredeneer eerst dat na 49 trekkingen nog één boon over is.
2. Kijk naar het netto effect op het aantal witte en zwarte bonen bij elk van de vier mogelijke trekkingen, en maak een tabel met vier kolommen: trekking, actie, stijging van het aantal zwarte bonen, en stijging van het aantal zwarte bonen.
3. Kijk naar de pariteit van het aantal witte bonen.

Opgave 1.10 (De ballenbak) In een doos zitten 100 rode, 100 witte en 100 blauwe ballen. Zolang dat mogelijk is, worden er drie ballen uit de doos gepakt. Als alle drie de ballen dezelfde kleur hebben, wordt er één van die drie teruggelegd. Als alle drie de ballen een verschillende kleur hebben, wordt een rode bal teruggelegd. Als er twee ballen dezelfde kleur hebben, wordt de afwijkende derde teruggelegd en bovendien een blauwe. Is het mogelijk te eindigen met een rode en een blauwe bal? (Hint: let op de rode ballen.)

Opgave 1.11 (MIU calculus) (Deze opgave is overgenomen uit [Hofstadter 1979]. Er is ook een aparte Wikipedia-pagina gewijd aan deze opgave.) Met de volgende regels kunnen symbool-rijtjes worden omgezet in andere symbool-rijtjes:

1. Achter een rijtje met een `I` aan het eind mag een `U` worden geplaatst;
2. Van het rijtje `Mx` mag `Mxx` worden gemaakt, voor elk rijtje symbolen `x`;
3. Elk voorkomen van `III` mag worden vervangen door een `U`;
4. Elk voorkomen van `UU` mag worden weggelaten.

Als voorbeeld bekijken we een aantal rijtjes die uit MI geproduceerd kunnen worden:

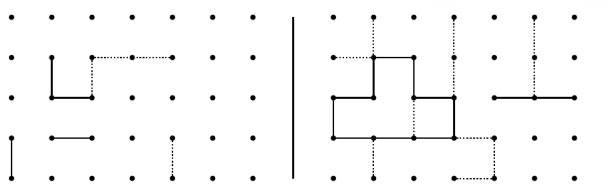
a.	`MI`	gegeven
b.	`MII`	uit *a.* volgens regel 2
c.	`MIIII`	uit *b.* volgens regel 2
d.	`MIIIIU`	uit *c.* volgens regel 1
e.	`MUIU`	uit *d.* volgens regel 3
f.	`MUIUUIU`	uit *e.* volgens regel 2
g.	`MUIIU`	uit *f.* volgens regel 4.

Vraag: is het mogelijk om beginnend met het rijtje `MI` uit te komen op het rijtje `MU`? Zo ja, hoe? Zo nee, waarom niet? (Hint: laat zien dat in geen enkel rijtje het aantal `I`'s deelbaar is door drie.)

Het belang van deze opgave is dat het een mini-voorloper is van de onvolledigheidsstelling van Gödel. De MIU-calculus kan worden opgevat als een bewijssysteem, en er wordt gevraagd de onbewijsbaarheid van `MU` aan te tonen.

Opgave 1.12 (Het lijnenspel) Het speelveld is ruitjespapier waarvan alleen de roosterpunten interessant zijn. Twee spelers, *A*(rie) en *B*(ert), doen om beurten een zet; *A* begint. Speler *A* doet een zet door twee

aangrenzende punten met een horizontale of een verticale lijn te verbinden, speler B doet hetzelfde, maar dan met stippellijnen. Spelers mogen reeds verbonden punten niet nog eens verbinden. Na vier zetten van beide spelers kan het bord er bijvoorbeeld uitzien als links in onderstaand plaatje.



Speler A wint als hij een gesloten figuur kan maken, zoals in het rechter plaatje; speler B heeft een andere opdracht: B wint als ze A kan verhinderen een gesloten figuur te maken. Vraag: is er voor één van beide spelers een winnende strategie? Zo ja, welke? (Hint: B kan A hinderen in het tekenen van zuid-west hoeken.)

Opgave 1.13 Piet ontwerpt een tile-based video game. Het idee is dat er een speelveld is dat kan worden bezet door (vierkante) tegels "bebouwing" of "kale grond". Meer soorten tegels zijn er niet. De bebouwing wordt niet beïnvloed door spelers maar door de game engine zelf. Elke tick is er een kleine kans dat een onbebouwde patch, die twee of meer bebouwde patches als buren heeft, zelf bebouwd raakt. Het speelveld is rechthoekig en er wordt met de Von Neumann omgeving gewerkt. (Zie hiervoor zo nodig Wikipedia.) De randen van het speelveld lopen niet in elkaar over; het speelveld is dus een echte rechthoek met hoeken en randen, en geen torus. Aan het begin van een spel-episode wordt de begin-bebouwing willekeurig gegenereerd. In het verloop van een episode kan elke onbebouwde patch in principe bebouwd raken. (Er is dus geen onbebouwbare grond zoals "water" of iets dergelijks.)

Het doel van de game-ontwerper is dat bebouwing kan toenemen, maar dat het ooit op een natuurlijke manier stopt, zodat nooit het hele veld bebouwd kan raken.

1. Beargumenteer dat zonder verdere maatregelen al aan deze eis is voldaan, mits er aan het begin van het spel niet al te veel bebouwd is. Laat dus zien dat bij een voldoende schaarse begin-bebouwing een veld nooit voor 100% bebouwd kan raken. (Hint: speel de animatie zelf eens na, en kijk wat er gebeurd met de totale bebouwingsomtrek.)

2. Geef een algemeen geldende maar niet-triviale bovengrens, N, voor de maximale begin-bebouwing. Je zoekt dus een redelijk groot getal, N, zo dat

   > $N =_{Def}$ Het aantal bebouwingstegels dat aan het begin van een episode mag worden geplaatst, zonder de mogelijkheid te riskeren dat een veld gedurende de animatie volloopt.

   Bijvoorbeeld: $N = 1$ is een bovengrens die werkt, maar deze bovengrens is triviaal ("flauw") want erg klein. Houd er rekening mee dat je bovengrens zal afhangen van de afmetingen van het veld: hoe groter het veld, hoe groter waarschijnlijk de bovengrens.

3. Beargumenteer dat een 10×10 veldje met 9 bebouwde tegels nooit vol kan lopen.

4. Beargumenteer dat een 10×10 veldje met 10 bebouwde tegels wel vol kan lopen. (Hint: geef een voorbeeld met een onfortuinlijke begin-situatie.)

5. Beargumenteer indien mogelijk dat de door jou gevonden bovengrens maximaal is. Dat wil zeggen dat bij elke grotere begin-bebouwing er wél een risico is dat het veldje volloopt.

Bewijs met een monovariant

Definitie 1.2 (Monovariant) *Een* monovariant *is een eigenschap die constant in één richting blijft veranderen tijdens het hele proces.*

Een voorbeeld van een monovariant is een tellertje die tijdens elke stap in het proces één ophoogt. Als bovendien bewezen kan worden dat dat tellertje niet boven een maximale waarde kan uitkomen, is in feite aangetoond dat het proces moet stoppen. Monovarianten worden dan ook vaak gebruikt om het stoppen van processen ("terminatie") aan te tonen.

Opgave 1.9 op blz. 8 bevat al een mooi voorbeeld van een monovariant, te weten het aantal bonen in de pot. Gemakkelijk is te beredeneren dat het aantal bonen in de pot na elke trekking met één daalt, zodat het proces uiteindelijk moet stoppen. Hier is nog een voorbeeld.

Voorbeeld 1.4 In een landhuis met 101 kamers lopen 999 zombies. Af en toe loopt een zombie naar een kamer met evenveel of meer zombies. In het onwaarschijnlijke geval dat alle zombies zich in één kamer bevinden, zal het landhuis imploderen tot een massieve grafsteen. Bewijs dat dit uiteindelijk zal gebeuren.

De aanpak is om een globale functie te definiëren die alleen maar kan stijgen als een zombie zich verplaatst, en die bovendien een maximale waarde kan aannemen. Eerst wat terminologie. Een *zombieconfiguratie* is een

beschrijving die aangeeft waar zombies zich bevinden. Laat C de verzameling van alle (zombie-) configuraties zijn. Ga na dat C eindig is. (Zombie 1 kan uit 101 kamers kiezen, zombie 2 kan uit 101 kamers kiezen, ..., zombie 999 kan uit 101 kamers kiezen.)

Definieer de functie

$$s : C \to \mathbb{N} : c \mapsto$$
som van kwadraten van alle aanwezigen per kamer

Dus de functie s van C naar $\mathbb{N}$ beeldt elke configuratie c af op een getal $s(c)$. Laten we dit de *spanning* noemen die in het landhuis heerst. Bijvoorbeeld de configuratie waarbij 5 kamers resp. 100, 200, 100, 500, en 99 zombies bevatten (en de overige kamers leeg zijn) levert een spanning op van $100^2 + 200^2 + 100^2 + 500^2 + 99^2$ eenheden.

Gemakkelijk is in te zien dat elke verplaatsing de spanning doet toenemen: als zombie x zich verplaatst van kamer A met m (inclusief x) aanwezigen naar een kamer B met $n \geq m$ aanwezigen, dan zal de spanning in kamer A afnemen met

$$m^2 - (m-1)^2 = 2m - 1$$

en in kamer B toenemen met $2n + 1$. Dus de totale spanning zal toenemen met

$$-2m + 1 + 2n + 1 = 2(n - m) + 2 \geq 2 \times 0 + 2 = 2.$$

Dus per verplaatsing zal de spanning met tenminste twee toenemen. De spanning is dus een monovariant. Omdat het aantal zombieconfiguraties eindig is, kan de spanning niet eindeloos toenemen. Op een gegeven moment bewegen zombies zich dus niet meer. De oorzaak daarvan kan alleen maar zijn dat geen enkele zombie nog een aanleiding heeft zich te verplaatsen. En dat kan alleen maar worden veroorzaakt doordat alle zombies zich al in een kamer bevinden met meer zombies dan in elke andere kamer. En dat laatste betekent weer dat alle zombies zich in één kamer moeten bevinden, wat op zijn beurt weer betekent dat het landhuis zal imploderen.

Opgave 1.14 In de senaat van Kirgizië heeft ieder lid ten hoogste drie vijanden. Vijandschap is wederzijds en niemand is z'n eigen vijand. Bewijs dat het mogelijk is de senaat in twee fracties te verdelen, zodanig dat in iedere fractie elk lid ten hoogste één vijand heeft. (Hint: maak twee willekeurige fracties en ruil.)

Bewijs door contrapositie

Een *contrapositie* van de bewering $P \Rightarrow Q$ ("als P dan Q") is de bewering $\neg Q \Rightarrow \neg P$ ("als niet-Q dan niet-P"). De volgende opgave leert je om te gaan met contraposities.

Opgave 1.15 Hier volgen een aantal beweringen. Doe voor elke bewering het volgende.

- Bepaal het universum. (De groep van individuen waarover de bewering wordt gedaan.) Het universum kun je liever te groot dan te klein kiezen.
- Bepaal of het om een implicatie gaat.

Als het om een implicatie gaat:

- Bepaal zo mogelijk of de implicatie geldt.
- Bepaal de contrapositie van de implicatie.
- Bepaal zo mogelijk of de contrapositie geldt.

1. Alle baarddragers zijn mannen. (Ook: als x een baarddrager is, dan is x een man.) De vraag om een geschikt universum wordt hier alvast voor je beantwoord: neem als universum: alle mensen die nu leven of ooit geleefd hebben.
2. Alle autorijders hebben vervoer.
3. Elk kind uit groep 5 zit op de basisschool.
4. Als een sporter is gediskwalificeerd, dan mag hij (zij) niet meedoen.
5. Heb je éénmaal op partij A gestemd, dan kun je niet meer op partij B stemmen.
6. Heb je op nog niet op partij A gestemd, dan kun je nog op partij B stemmen.
7. Als je zwangerschapstest positief is, dan ben je in verwachting.
8. Als je zwangerschapstest negatief is, dan is er toch nog een mogelijkheid dat je in verwachting kunt zijn.

De volgende opgave gaat ook met contrapositie, maar is wat moeilijker.

Opgave 1.16 Bewijs door contrapositie: als $n \in \mathbb{N}$ geen kwadraat is, dan is $\sqrt{n} \notin \mathbb{Q}$. (Hint: neem aan dat eventuele breuken vereenvoudigd zijn.)

Sommige beweringen laten zich moeilijk direct bewijzen:

In een groep van 50 *mensen zijn er altijd* 4 *die in dezelfde maand jarig zijn.* (5)

Hoe pakken we dat aan? Welke maand is dat dan? Laten we (5) weergeven als een implicatie

$$P \Rightarrow Q \tag{6}$$

waarbij

P = "de groep bestaat uit 50 mensen";

Q = "er zijn altijd 4 of meer mensen in dezelfde maand jarig"

De contrapositie van (6), is hiermee equivalent:

$$\text{niet}Q \Rightarrow \text{niet}P$$

De contrapositie is makkelijker te bewijzen:

Stel, er zijn nooit 4 of meer mensen in dezelfde maand jarig. Dat impliceert dat er ten hoogste 3 mensen in januari jarig zijn, ten hoogste 3 mensen in februari, ..., en ten hoogste 3 mensen in december. We hebben nu $12 \times 3 = 36$ mensen verdeeld over 12 maanden. Persoon 37 kan nergens meer bij, want alle groepjes van elke maand zitten al vol. Dus als er nooit 4 of meer mensen in dezelfde maand jarig zijn, kan de groep uit ten hoogste 36 personen bestaan, en $36 < 50$. □

Opgave 1.17 Bewijs of geef een tegenvoorbeeld:

1. In een groep van 50 mensen zijn er altijd 5 of meer in dezelfde maand jarig.

2. In een groep van 50 mensen zijn er altijd 6 of meer in dezelfde maand jarig.

Opgave 1.18 Het bewijs van ongelijkheid (4) werd ter oefening aan de lezer overgelaten, maar is nog niet zo makkelijk. Deze opgave hoopt je naar het bewijs te leiden. Stel $0 \le x \le y$. Bewijs de volgende onderdelen.

1. $0 \le x^2 \le xy$. (Gebruik dat $x \ge 0$.)
2. $0 \le xy \le y^2$.
3. $x^2 \le y^2$.
4. Als $x^2 \le y^2$, dan $x \le y$. (Hint: gebruik $x^2 \le y^2 \Leftrightarrow y^2 \not< x^2$ en contrapositie.)

Bewijs uit het ongerijmde

Soms kan een uitspraak alleen maar worden bewezen door aan te nemen dat deze onwaar is, met als doel te laten zien dat zo'n aanname onvermijdelijk leidt tot een tegenspraak.[5]

Voorbeeld 1.5 Bewijs uit het ongerijmde dat er geen kleinste geheel getal bestaat.

Stel dat we dit direct zouden willen bewijzen. Dan zouden we de afwezigheid van een kleinste geheel getal moeten aantonen door een direct argument. Dit directe argument zou dan zoiets kunnen zijn als: "Ik heb heel lang gezocht naar een kleinste geheel getal, maar ik kon hem niet echt vinden. En ik heb echt heel hard gezocht." Dit is geen bewijs dat een kleinste geheel getal niet bestaat. Een kritische toehoorder kan bijvoorbeeld tegenwerpen dat een kleinste geheel getal toch kan bestaan omdat je niet goed genoeg hebt gezocht.

We geven nu een bewijs uit het ongerijmde.

Bewijs: Stel dat er toch een kleinste geheel getal zou bestaan. Noem dit n. Omdat n een geheel getal is, is $n-1$ ook een geheel getal. Maar $n-1 < n$, wat is tegenspraak is met de aanname dat n het kleinste gehele getal zou zijn.

De aanname dat er toch een kleinste geheel getal bestaat leidt dus tot een tegenspraak. Deze aanname kan dus niet waar zijn, dus kan er geen kleinste gehele getal bestaan. □

Opmerkingen:

1. Een mnemonic voor een bewijs uit het ongerijmde constructie is: "stel niet, dan toch".

2. De Latijnse benaming voor de notie "bewijs uit het ongerijmde" is *reductio ad absurdum*.

3. Bij een bewijs uit het ongerijmde moeten alle andere aannamen waar zijn, en moet de redenering die leidt naar de tegenspraak kloppen. Anders kan de tegenspraak ook worden veroorzaakt door een ondeugdelijke redenering of door andere mogelijk foute aannamen.

4. Een bewijs uit het ongerijmde wordt alleen gegeven als een constructief bewijs niet voorhanden is, of als een constructief bewijs langer is. Het geven van een direct bewijs heeft dus altijd de voorkeur.

5. Een bewijs uit het ongerijmde is een bijzonder geval van een bewijs door contrapositie. De bewering Q kan namelijk geschreven worden als

   $$true \Rightarrow Q$$

   ("als [lege conditie] dan Q"). De contrapositie hiervan is $\neg Q \Rightarrow \neg true$. Oftewel: $\neg Q \Rightarrow false$, ofwel: "als niet-Q dan tegenspraak".

Het meest bekende bewijs uit het ongerijmde is het bewijs dat wortel twee geen breuk is: $\sqrt{2} \notin \mathbb{Q}$. Dit werd ongeveer vijf eeuwen voor Christus ontdekt door de Grieken, die hier behoorlijk van onderstboven waren. Zij dachten namelijk tot dan toe dat alle lengte-eenheden als breuken kunnen worden uitgedrukt. (De Grieken waren overigens onbekend met symbolen als "$\sqrt{}$," "$\mathbb{Q}$," en "$\in$".)

Bewijs: Stel dat wortel twee toch een breuk is. We schrijven

$$\sqrt{2} = \frac{n}{m},$$

waarbij $n, m \in \mathbb{Z}$. Zonder beperking der algemeenheid, i.e., zonder iets aan de algemeenheid van het bewijs af te doen, mogen we aannemen dat $n, m \in \mathbb{N}$, en mogen we aannemen dat deze breuk zo veel mogelijk is vereenvoudigd. (Mee eens?)

Beide kanten kwadrateren en vermenigvuldigen met m^2 geeft

$$2m^2 = n^2.$$

Dus n^2 is even. Nu gebruiken we het eenvoudig te bewijzen feit dat, als een getal even is, zijn kwadraat dat ook is, en omgekeerd. Dus n is even. Schrijf $n = 2k$. Dus $n^2 = 4k^2$, en dus

$$2m^2 = 4k^2.$$

Maar dan is m ook even, en was de oorspronkelijke breuk klaarblijkelijk toch niet vereenvoudigd. Tegenspraak. □

Opgave 1.19 Bewijs dat, als een getal even is, zijn kwadraat dat ook is. Bewijs ook dat, als een kwadraat even is, het oorspronkelijke getal dat ook is. (Hint: contrapositie.)

Opmerkingen:

– Als we teller en noemer door twee delen en verder vereenvoudigen, kan de redenering worden herhaald. Dus een breuk gelijk aan wortel twee bestaat echt niet.

[5] In feite is dit een filosofische uitspraak. Veel wiskundigen zouden het met deze uitspraak eens zijn. Constructivistisch wiskundigen niet. We gaan hier niet verder op in.

– Er bestaan constructieve bewijzen van $\sqrt{2} \notin \mathbb{Q}$, maar deze zijn minder direct.

Opgave 1.20 Bewijs de volgende uitspraken uit het ongerijmde:

1. Er is geen kleinste reëel getal.
2. Het is onmogelijk om oneven getallen te schrijven als de som van drie even getallen.
3. De som van een rationaal getal (breuk) en een irrationaal getal (reëel getal dat niet als breuk te schrijven is), is irrationaal.
4. Een Diofantische vergelijking is het gelijk aan nul stellen van een veelterm met meerdere *geheeltallige* variabelen. Bewijs dat de Diofantische vergelijking
$$x^2 - y^2 - 1 = 0$$
geen positive oplossingen heeft.
5. De vergelijking
$$x^3 + x + 1 = 0$$
heeft geen rationele oplossingen. (Hint: schrijf x als breuk, maak een Diofantische vergelijking, en onderscheid gevallen waarbij variabelen even dan wel oneven zijn.)
6. Er bestaan oneindig veel priemgetallen. (Hint: Vorm het product van alle priemgetallen + 1, en bekijk een priemdeler.)

Bewijs door volledige inductie

Volledige inductie is een veelvoorkomende bewijstechniek op $\mathbb{N}$ of $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Het zegt dat als een eigenschap waar is voor het eerste getal (1 resp. 0), en die eigenschap plant zich van getal tot getal voort, dan is die eigenschap voor alle getallen waar.

Voorbeeld 1.6 Bewijs met volledige inductie dat

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}. \tag{7}$$

Laten we het eerst zonder volledige inductie doen :-). Dit directe bewijs voor $n = 100$ schijnt door de wiskundige Gauss (1777-1855) te zijn gegeven toen hij als kind op de basisschool met dit probleem werd geconfronteerd. Het scheen naar verluid als som aan de klas te zijn voorgelegd. Als n even is, bijvoorbeeld $n = 100$, dan

$$\underbrace{\begin{array}{cccccc} 1 & 2 & 3 & \ldots & 99 & 100 \\ 100 & 99 & 98 & \ldots & 2 & 1 \\ \hline 101 & 101 & 101 & \ldots & 101 & 101 \end{array}}_{50 \text{ keer}} +$$

Als n oneven is, werkt dit niet meer, althans het wordt gecompliceerder. Wat wel kan voor iedere n is gewoon doortellen

$$\underbrace{\begin{array}{cccccc} 1 & 2 & 3 & \ldots & n-1 & n \\ n & (n-1) & n-2 & \ldots & 2 & 1 \\ \hline n+1 & n+1 & n+1 & \ldots & n+1 & n+1 \end{array}}_{n \text{ keer}} +$$

we hebben dan twee keer zo veel, maar dan deel je daarna gewoon weer door twee. Klaar.

Nu met volledige inductie. We moeten eerst laten zien dat de eigenschap waar is voor het eerste getal, zo uit de formule af te lezen is dat eerste getal gelijk aan 1. Welnu, $1 = 1(1+1)/2$ dus dat zit goed. De inductie-basis is gelegd.

Nu moeten we laten zien dat (7) een eigenschap is die zich voortplant over alle getallen na 1. Laten we aannemen dat de eigenschap zich heeft voortgeplant tot en met $n-1$. Dus

$$1 + 2 + \cdots + (n-1) = \frac{(n-1)n}{2}. \tag{8}$$

Dit is de inductie-hypothese. We moeten nu laten zien dat (7). Dat kan zo:

$$\begin{aligned} 1 + 2 + \cdots + n &= (1 + 2 + \cdots + n - 1) + n \\ &= \frac{(n-1)n}{2} + n \qquad (!) \\ &= \frac{(n-1)n}{2} + \frac{2n}{2} \\ &= \frac{n(n+1)}{2}. \end{aligned}$$

Op de plek van het uitroepteken is de inductie-hypothese (8) toegepast. □

In het bewijs hierboven werd de inductie-stap gemaakt van $n-1$ naar n. Maar je mag hem ook maken van n naar $n+1$, immers n is maar een variabele.

Opgave 1.21 Bewijs (7) met een inductie-stap van n naar $n+1$. Hoe ziet je inductie-hypothese er uit?

Opgave 1.22 Bewijs met volledige inductie. Geef de inductie-basis aan, en de inductie-hypothese. (Alle $n \in \mathbb{N}_0 =_{Def} \mathbb{N} \cup \{0\}$ tenzij anders vermeld.)

1. $2^n \leq 3^n$
2. $1 + 3 + 5 + \cdots + (2n-1) = n^2$, $n \geq 1$
3. $1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
4. $1^2 + 2^2 + 3^2 + \cdots + n^2 = n(n+1)(2n+1)/6$
5. $n^2 \geq 2n + 1$, voor $n \geq 3$
6. $n! \geq 2^n$, voor $n \geq 4$
7. $3 \mid (n^3 + 2n)$ (De notatie $k|m$ betekent: k deelt m.)
8. $(r+1)^n \geq nr + 1$, voor $r > -1$ reëel. Laat ook zien dat $r > -1$ een noodzakelijke voorwaarde is.

Bij *zwakke inductie* gebruik je alleen de aanname dat de eigenschap geldt voor het voorlaatste getal. Bij *sterke inductie* gebruik je de aanname dat de eigenschap geldt voor alle kleinere getallen. Beide vormen van inductie zijn gelijkwaardig, maar in sommige bewijzen kun je alleen sterke inductie gebruiken.

Voorbeeld 1.7 Elk getal $n \geq 2$ kan worden ontbonden in priemgetallen, en zo'n ontbinding is uniek.

Bewijs: Elk getal $n \geq 2$ bezit een kleinste factor p. Deze factor p is uniek, want twee verschillende factoren kunnen niet beiden de kleinste zijn.

Elke ontbinding van n bevat noodzakelijk die kleinste factor p. Stel immers dat $n = p_1 \dots p_k$ een ontbinding zou zijn met factoren op volgorde van grootte en $p_1 > p$. Omdat $p < p_1$ zou dan gelden $n' =_{Def} p \cdot p_2 \dots p_k < n$. Uit $p \mid n$ en $p \mid n'$ zou dan volgen $p \mid (n - n')$ wat hetzelfde is als $p \mid (p_1 - p)p_2 \dots p_k$. Omdat $p \nmid (p_1 - p)$ zou dan $p \mid p_2 \dots p_k$ en dus zou p dan tóch één van de $p_2, \dots, p_k$ moeten zijn. Dat kan niet, want $p < p_2, \dots, p_k$. Tegenspraak. Dus bestaat er geen ontbinding die de kleinste factor vermijdt. Dus elke ontbinding bevat de kleinste factor.

Nu terug naar het bewijs van de oorspronkelijke stelling. Als inductie-basis nemen we $n = 2$. Ga na dat de stelling waar is voor $n = 2$. Nu de inductiestap. Stel $n > 2$. Als inductie-hypothese nemen we aan dat de stelling waar is voor alle k met $2 \leq k < n$. Inmiddels weten we dat elke ontbinding van n de (unieke) kleinste factor p moet bevatten. Dus elke ontbinding van n kan, nee moet, geschreven worden als $n = p \cdot m$, waarbij per sterke inductie de ontbinding van m ook uniek is. Hiermee is de ontbinding van n dus ook uniek. □

We gaan dit resultaat gebruiken bij Gödelnummeringen, zie blz. 60.

Opgave 1.23 1. Een chocoladetablet van $m \times n$ blokjes moet volledig worden opgebroken. Bewijs dat je tenminste $mn - 1$ keer moet breken.

2. Gegeven zijn getallen $b_1, b_2, \dots$, zó dat $b_1 = 1$, $b_2 = 2$ en

$$b_{n+2} = b_{n+1} + 2b_n$$

voor $n \geq 1$.

   (a) Geef een directe formule voor b_n. (Bijvoorbeeld: $b_n = 2^n - 3$.)

   (b) Bewijs met sterke inductie dat de formule correct is.

3. Elk bedrag boven de 11 Euro kan worden geplakt met postzegels van 4 en 5 Euro.

   Bewijs dit feit één keer met zwakke inductie, en één keer met sterke inductie. (Hint: sterke inductie verloopt makkelijker maar heeft, om te kunnen opstarten, vier basisgevallen nodig.)

Dan is er nog *structurele inductie*. Deze wordt meestal toegepast om eigenschappen van recursief opgebouwde structuren te bewijzen. Het principe van structurele inductie zegt dat als een eigenschap waar is voor de kleinste structuur, en die eigenschap zich voortplant als de structuur in stapjes wordt opgebouwd, dan is die eigenschap voor alle structuren waar.

Voorbeeld 1.8 (Rekenkundige expressies) Laat R de taal zijn van eenvoudige rekenkundige expressies, kortweg *expressies*.

1. Elk geheel getal is een expressie.
2. Elke hoofdletter is een expressie.
3. Als φ een expressie is, dan is $(-\varphi)$ ook een expressie.
4. Als φ en ψ expressies zijn, dan is $(\varphi + \psi)$ ook een expressie.
5. Als φ en ψ expressies zijn, dan is $(\varphi \times \psi)$ ook een expressie.
6. Geen enkel ander ding is een rekenkundige expressie.

Opmerkingen:

1. Op deze manier kun je oneindig veel verschillende expressies maken!
2. We gebruiken Griekse letters zoals φ (phi) en ψ (psi) om geen verwarring te krijgen met expressies van de vorm $A, B, \dots$.
3. We missen regels om expressies te maken voor aftrekken en deling, maar voor deze opgave is dat niet zo belangrijk.
4. Expressies die niet zijn te ontleden in deelexpressies worden *atomen* genoemd. In dit geval zijn atomen dus getallen en hoofdletters.
5. Regel (6) doet misschien en beetje streng aan. Maar als die regel er niet staat, laat bovenstaande definitie de deur open voor de mogelijkheid dat uitdrukkingen, anders dan die gedefinieerd in (1)-(5), ook rekenkundige expressies kunnen zijn.

Opgave 1.24 Bepaal van de volgende expressies of het elementen van de taal R zijn. Let op: haakjes mogen niet zo maar worden weggelaten.

A, -5, (-6), $(3+4)$, $(3-4)$, $3+4$, $3+(F+4)$, $(3 \times (G+4))$, $(-(K+(3+A)))$, $-(P+(P+4))$

Voorbeeld 1.9 We gaan met structurele inductie bewijzen dat elke expressie uit R een even aantal haakjes bevat.

Bewijs: Zij $\varphi \in R$. We weten niet hoe φ er uit ziet. Wat we wel weten is dat volgens Clausule (6) φ maar op vijf verschillende manieren kan zijn ontstaan: volgens (1), volgens (2), volgens (3), volgens (4), of volgens (5).

- Als φ volgens (1) gemaakt is, is het een geheel getal. Een getal bevat nul haakjes, en nul is even.

- Als φ volgens (2) gemaakt is, is het een hoofdletter. Een hoofdletter bevat nul haakjes.
- Als φ volgens (3) gemaakt is, is het van de vorm $(-\psi)$. Omdat ψ echt minder symbolen dan φ bevat, mogen we aannemen dat ψ een even aantal haakjes bevat. De expressie $\varphi = (-\psi)$ bevat dan ook een even aantal haakjes, want:

  even + 2 = even.
- Als φ volgens (4) gemaakt is, is het van de vorm $(\psi \times \chi)$. (We hebben een derde Griekse letter nodig, χ, geschreven: chi, spreek uit: "gi".) Omdat ψ en χ echt minder symbolen dan φ bevatten, mogen we van beiden aannemen dat ze een even aantal haakjes bevatten. De expressie $\varphi = (\psi \times \chi)$ bevat dan ook een even aantal haakjes, want:

  even + even + 2 = even.
- Het bewijs van geval (5) verloopt analoog aan het bewijs van geval (4).

Met (6) kunnen we nu concluderen dat voor elk type expressie (atomair, minus, optelling en vermenigvuldiging) bewezen is dat deze een even aantal haakjes bevat. □

Structurele inductie is niet beter of anders dan 'normale" volledige inductie. In feite is structurele inductie gelijk aan normale volledige inductie waarbij het getal waarop we inductie plegen gelijk is aan de grootte van de onderliggende structuren, in ons geval rekenkundige expressies.

In de volgende opgave gaan we met structurele inductie bewijzen dat het aantal karakters in een expressie, inclusief haakjes en operatoren, kortweg de *lengte* van een expressie, *lineair* afhangt van het aantal operatoren in de expressie. Dit is een belangrijk resultaat dat onder meer gebruikt wordt in automatisch redeneren. Op deze manier kan namelijk worden gewaarborgd dat complexe expressies met veel operatoren niet veel te lang worden om te worden herkend door bijvoorbeeld automatische stellingenbewijzers.

Opgave 1.25 We spreken af dat de *lengte* van een atoom gelijk is aan 1. (Zie opmerking 4 op blz. 13.)

1. Bouw stapje voor stapje een expressie op met alleen binaire operatoren + en ×, bijvoorbeeld:

   $$P \to (4+P) \to (A \times (4+P)) \to ((A \times (4+P))+K) \to \ldots$$

   Vul de volgende tabel in:

aantal operatoren:	0	1	2	3	4	5	...
expressie-lengte:							...

   (Ter contrôle: met 7 binaire operatoren bestaat een expressie uit 29 karakters; met 8 uit 33.)
2. Stel een formule op die aangeeft hoe de lengte van een expressie φ die alleen maar is opgebouwd uit binaire operatoren, afhangt van het aantal operatoren. Voorbeeld:

   $$l(\varphi) = 5o(\varphi) - 3. \tag{9}$$

   (De 5 en de −3 zijn fout.) Je hoeft de geldigheid van de formule nog niet te bewijzen.
3. Doe onderdelen 1 en 2 voor expressies met alleen unaire operatoren.
4. Doe onderdelen 1 en 2 voor expressies met zowel unaire als binaire operatoren. (Hint: je krijgt iets als formule (9) maar dan in de vorm van een ongelijkheid.)
5. Bewijs de het in onderdeel 4 gevonden antwoord met inductie naar de opbouw van expressies.
6. Zij R' de verzameling van expressies die ontstaat uit R als we, zonder verlies van betekenis, haakjes weg mogen laten. Geef een formule voor de maximale lengte van expressies, uitgedrukt in het aantal gebruikte operatoren.
7. Zij R'' de verzameling van expressies die ontstaat uit R als we (alleen) de buitenste haakjes weg mogen laten. Dezelfde vraag.

Opgave 1.26 Wat is er fout aan het volgende pseudo-inductieargument?

Bewering: *alle mensen hebben dezelfde kleur ogen.*

Bewijs: met inductie. Inductie-basis: een groep bestaande uit één persoon heeft dezelfde kleur ogen, dus dat zit goed. Inductie-hypothese: alle groepen van $n-1$ mensen hebben dezelfde kleur ogen. Neem nu een groep van n mensen:

$$A = \{a_1, a_2, a_3, \ldots, a_{n-1}, a_n\}.$$

Haal eerst persoon a_n uit A. De resterende groep $B = \{a_1, a_2, a_3, \ldots, a_{n-1}\}$ bezit per inductie dezelfde kleur ogen. Haal nu persoon a_{n-1} uit A. De resterende groep $C = \{a_1, a_2, a_3, \ldots, a_{n-2}, a_n\}$ bezit per inductie dezelfde kleur ogen. Omdat bijvoorbeeld persoon a_1 in zowel B als C zit, moet iedereen in $B \cup C = A$ dezelfde kleur ogen bezitten.

Opgave 1.27 Geef je mening over het volgende argument.

Bewering: *alle natuurlijke getallen zijn interessant.*

Bewijs: stel niet. Dan is er een kleinste getal, u, dat niet interessant is. Maar precies dat feit maakt u interessant! Tegenspraak. □

Niet-constructieve bewijzen

Niet-constructieve bewijzen zijn een beetje gek. Je bewijst het bestaan van een ding (of dingen) met een bepaalde eigenschap, zonder dat ding (of die dingen) ook daadwerkelijk te construeren.

Stelling 1.1 *Er bestaan getallen $a, b \notin \mathbb{Q}$, zo dat $a^b \in \mathbb{Q}$.*

Bewijs: We gaan $q = \sqrt{2}^{\sqrt{2}}$ bekijken. Nu zijn er twee gevallen mogelijk: $q \in \mathbb{Q}$ of $q \notin \mathbb{Q}$. Als $q \in \mathbb{Q}$ zijn we klaar, immers $a = b = \sqrt{2}$ voldoen. Als $q \notin \mathbb{Q}$, neem dan $a = q$ en b nog steeds $\sqrt{2}$. Dan

$$a^b = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{(\sqrt{2}\cdot\sqrt{2})} = \sqrt{2}^2 = 2$$

en $2 \in \mathbb{Q}$. □

Het bijzondere aan dit bewijs is dat het geen uitsluitsel geeft over de rol van q, en ook niet zegt voor welke getallen a en b de stelling nu eigenlijk geldt!

Veel non-constructieve bewijzen in de verzamelingenleer verlopen als volgt: er wordt beweerd dat er een ding x bestaat die wel aan eigenschap P maar niet aan eigenschap Q voldoet. Een niet-constructief bewijs voor het bestaan van zo'n individu x laat dan zien dat de verzameling van alle dingen met eigenschap Q echt meer elementen bevat dan de verzameling van alle dingen met eigenschap P.

Veel non-constructieve bewijzen in de logica verlopen als volgt: er wordt beweerd dat, als P geldt, dan bestaat er een constructie voor object x (een recept om x te bouwen). Vaak word dan de contra-positie bewezen: als x niet kan worden geconstrueerd, dan kan P ook niet waar zijn. Op deze manier is er dus een constructiebewijs gegeven zonder dat er ooit iets is geconstrueerd!

Nogal wat non-constructieve bewijzen zijn niet mogelijk zonder het zogenaamde keuzeaxioma te accepteren (blz. 51).

Opgave 1.28 Bewijs de volgende beweringen. Leg uit welk non-constructief aspect je bewijs bevat.

1. (Pigeon-hole principle) Als je n voorwerpen in k dozen stopt, en $k < n$, dan is er een doos waar twee of meer voorwerpen in zitten. (Hint: bewijs de contrapositie.)
2. Elk deelbaar getal n bezit een deler kleiner of gelijk aan $\sqrt{n}$.
3. Elke veelterm $a_n x^n + \cdots + a_1 x + a_0$ met n oneven, bezit een oplossing in $\mathbb{R}$. (Hint: de grafiek van zo'n veelterm is continu, verdwijnt rechts naar oneindig en links naar min oneindig.)
4. Elke uitdrukking $n^3 - n$ is deelbaar door drie. (Hint: ontbind in factoren.)
5. In een groep van zes mensen is een groepje van 3 dat elkaar allemaal kent, of een groepje van 3 dat elkaar juist allemaal niet kent.

Uniciteitsbewijzen

Soms wordt je gevraagd te bewijzen dat er maar één ding met een bepaalde eigenschap bestaat.

Voorbeeld 1.10 Er bestaat maar één element $0 \in \mathbb{Z}$ zo dat voor alle $k \in \mathbb{Z}$ geldt dat $k + 0 = k$.

Bewijs: Stel er is een element dat de taak van nul op zich neemt, i.e., stel er is een $n \in \mathbb{Z}$, zó dat voor alle $n \in \mathbb{Z}$ óók geldt dat $k + n = k$. We gaan laten zien dat n niet anders dan gelijk moet zijn aan 0:

$$n = n + 0 = 0 + n = 0.$$

De eerste gelijkheid volgt uit de nul-eigenschap van 0 ($k + 0 = k$, voor iedere k, dus ook voor $k = n$), de tweede gelijkheid volgt uit de commutativiteit van optelling ($a + b = b + a$), de derde gelijkheid volgt uit de nul-eigenschap van n ($k + n = k$, voor iedere k, dus ook voor $k = 0$). □

Het is misschien moeilijk vat te krijgen op dit bewijs omdat verwezen wordt naar regels waarvan de waarheid voorheen stilzwijgend werd aangenomen. Het helpt als je bij jezelf nagaat welke regels dat zijn, en hoe die regels zijn geïnstantieerd (ingevuld).

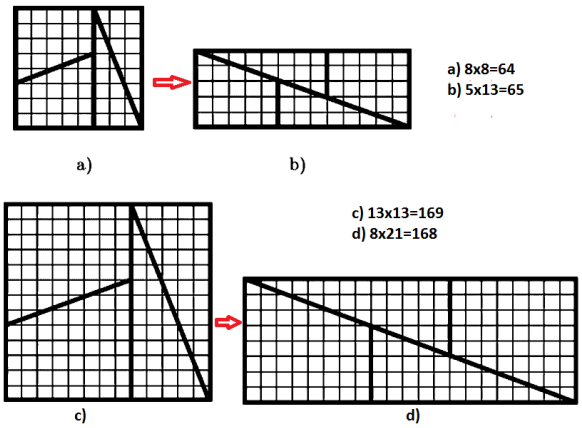
Behalve bewijs door gevalsonderscheid, bewijs door contrapositie, bewijs door volledige inductie, niet-constructieve bewijzen, en uniciteitsbewijzen zijn er nog andere bewijstechnieken, zoals equivalentiebewijzen, bewijs door herhaling, bewijs door plaatjes kijken, bewijs door semantische verschuiving, bewijs door gesticulatie, bewijs door charisma, bewijs door afmatting, bewijs door uitputting, bewijs door vertroebeling, bewijs door autoriteit, bewijs door intimidatie, bewijs door op de man te spelen, bewijs door onbegrijpelijke notatie, bewijs door steeds harder praten, bewijs door reductie naar het verkeerde probleem, bewijs door verwijzing naar ontoegankelijke literatuur, bewijs door inspraak ("wat vind jij?"), en bewijs door democratie (de meeste stemmen gelden). Een bespreking van deze technieken valt buiten het bestek van deze tekst.

1.7 Foute bewijzen

Op dit punt aangekomen zal het je wellicht niet verbazen dat er talloze manieren zijn waarop een bewijs kan ontsporen. We laten niet alle manieren zien maar lichten er een paar uit.

Plaatje

Soms merkt men op: "dit hoef je toch niet te bewijzen, dat kun je toch zo wel zien uit het plaatje"? Voor deze personen hebben we het volgende "bewijs" opgeduikeld. Dit "bewijs", of beter: pseudo-bewijs, laat zogenaamd zien dat een vierkant groter kan worden gemaakt door het op een bepaalde manier te verknippen en weer aan elkaar te plakken:



Het tweede plaatje laat zien dat het door middel van deze techniek ook mogelijk is een vierkant te *verkleinen* tot een rechthoek met een oppervlakte die kleiner is dan de oppervlakte van het oorspronkelijke vierkant.

Opgave 1.29 Kun je aangeven wat er fout gaat in dit "bewijs"?

Pseudo-bewijzen die gebaseerd zijn op plaatjes, zijn talrijk. De fout bij dergelijke bewijzen schuilt in het feit dat de opgevoerde plaatjes een foute voorstelling van zaken geven of niet alle mogelijke situaties of "liggingen" representeren.

Brulaap

Vervelend is wanneer iemand iets "bewijst" dat op zich waar is, maar waarvan het argument zelf niet deugt. Het trekken van een juiste conclusie op basis van een ondeugdelijk bewijs wordt in het Engels een "howler" (Ned.: "brulaap") genoemd.

Zie het volgende "bewijs" van het feit dat 16/64 kan worden vereenvoudigd tot 1/4:

$$\frac{16}{64} = \frac{1\cancel{6}}{\cancel{6}4} = \frac{1}{4}.$$

Het vervelende van een dergelijke manoeuvre is dat iemand terecht kan beweren dat wat hij afleidt waar is, maar onterecht daaruit concludeert dat zijn argument "dus" klopt. In dit geval, met het wegstrepen van de zessen, is het makkelijk de fout aan te wijzen. In realistische (en dus meestal ingewikkeldere) gevallen kan het moeilijk zijn de vinger op de zere plek te leggen.

Oneindige som

Jan-Willem beweert:

$$1-1+1-1+\cdots = 0.$$

Zijn bewijs:

$$\begin{aligned} 1-1+1-1+\cdots &= \\ &= (1-1)+(1-1)+\cdots = 0+0+\cdots = 0. \end{aligned}$$

Marlies beweert:

$$1-1+1-1+\cdots = 1.$$

Haar bewijs:

$$\begin{aligned} 1-1+1-1+\cdots &= \\ &= 1-(1-1)-(1-1)-\cdots = 1-0-0-\cdots = 1. \end{aligned}$$

Edith beweert:

$$1-1+1-1+\cdots = \frac{1}{2}.$$

Haar bewijs:

$$\begin{aligned} S = 1-1+1-1+\cdots &= \\ &= 1-(1-1+1-1+\ldots) = 1-S. \end{aligned}$$

De variabele S oplossen geeft $S = 1/2$.

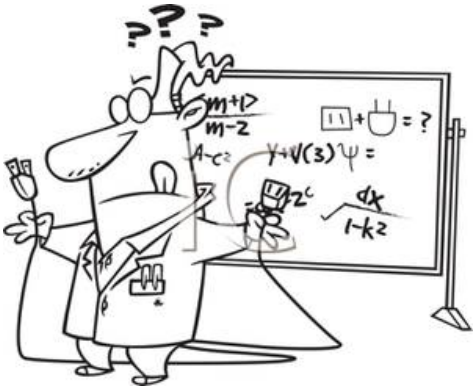
Wat is hier aan de hand? Al in de 17e eeuw braken vooraanstaande wiskundigen zich het hoofd over oneindige sommen. Gottfried Wilhelm Leibniz (1646-1716), bijvoorbeeld, een eminent wiskundige, filosoof, logicus, natuurkundige, historicus, rechtsgeleerde en diplomaat, en samen met Newton de grondlegger van de differentiaal- en integraalrekening, was bijvoorbeeld van mening dat de uitkomst gelijk aan 1/2 moest zijn omdat dit het gemiddelde is van de uitkomsten die Jan-Willem en Marlies vonden. (Jan-Willem en Marlies leefden toen nog niet, maar je begrijpt wat we bedoelen.) Of hij bekend was met Edith's argument (dat inderdaad op 1/2 uitkomt) weten we niet.

De oneindige som is volgens de moderne wiskunde gedefinieerd als de limiet van de deelsommen. De eerste deelsom is 1; de tweede deelsom is $1-1=0$; de derde deelsom is $1-1+1=1$; de vierde deelsom is $1-1+1-1=0$. In het algemeen is de n^e deelsom gelijk aan 0 voor even n, en gelijk 1 voor oneven n. De rij deelsommen is dus gelijk aan $1,0,1,0,1,0,\ldots$ en bezit dus geen limiet, i.e., de rij deelsommen convergeert dus niet naar een vast getal. De oneindige som $1-1+1-1+\ldots$ is dus volgens de moderne wiskunde niet gedefinieerd. Volgens de moderne wiskunde bestaat deze dus niet.

Krukken

Als laatste behandelen we niet zozeer een categorie foute bewijzen als wel een categorie wiskundebeoefenaars, genaamd krukken.

Een kruk (Eng.: "crank", "crackpot," "crook") is een kleinerende benaming voor iemand die met geestdrift vasthoudt aan het bewijs van een wiskundige bewering, waarvan meestal al lang geleden is komen vast te staan dat deze niet waar is, of waarvan uiterst onwaarschijnlijk is dat deze waar is.



Met name de laatste mogelijkheid geeft de crank altijd een sprankje hoop en, in zijn ogen (de crank is meestal een man), het recht om professionele wiskundigen, logici, of informatici met zijn theorieën te blijven bestoken.

Een crank zal typisch alle aangevoerde tegenargumenten terzijde schuiven, meestal door op te merken dat men zijn bewijs beter moet lezen, of dat men hem niet heeft begrepen. (Soms gevolgd door een verwijzing naar andere beroemde wiskundigen die “eerst ook niet werden begrepen”.) Typisch voor cranks is ook dat zij zelden of nooit fouten toegeven, zelfs als het een triviale fout betreft. In discussie gaan met een crank heeft daarom meestal weinig zin.

Voorbeelden:

- De bewering Goldbach’s vermoeden te hebben bewezen.

  Op 7 juni 1742 schreef de Brandenburgs-Pruisische wiskundige Christian Goldbach een brief naar de op dat moment vermaarde wiskundige Leonhard Euler, waarin Goldbach het vermoedde uitte dat elk even getal groter dan 2 geschreven kan worden als de som van twee priemgetallen. Veel cranks beweren dit vermoeden te hebben bewezen. Tot op heden[6] blijken alle geclaimde bewijzen ondeugdelijk of onbegrijpelijk.

- De bewering het priemtweelingvermoeden te hebben opgelost. (Lees verder op blz. 34.).

- De bewering het vermoeden van Collatz te hebben bewezen. (Lees verder op blz. 42 en 214.)

- De ontkenning van Cantor’s diagonaalproces. (Lees verder op blz. 49.)

- De bewering de laatste stelling van Fermat te hebben opgelost. (Lees verder op blz. 67.)

- De bewering een computer-programma te hebben geschreven dat kan controleren of programma’s in een bepaalde taal, bijvoorbeeld C, stoppen. (Lees verder op blz. 215.)

- De bewering het $P = NP$ probleem te hebben opgelost. Dit probleem werd in 1971 geformuleerd door Amerikaans-Canadees informaticus S.A. Cook (1939), en is op dit moment[7] zonder twijfel nog steeds het meest belangrijke probleem in de informatica.

  Het vereist enige voorkennis en tijd van de lezer om de vraag $P =_? NP$ netjes uitgelegd te krijgen. Op dit punt volstaan we met te vermelden dat het Clay Mathematics Institute in 2000 heeft aangekondigd $1,000,000$ dollar uit te reiken aan diegene die het $P = NP$ probleem kan bewijzen of weerleggen. Tot nu toe[8] heeft niemand zich gemeld.

De geschiedenis wijst uit dat cranks niet per sé amateurs hoeven te zijn. Er zijn gevallen bekend waarin hoogleraren van naam en faam vervielen tot crankdom. Denk aan de hoogleraar natuurkunde die er op stond dat zijn werk om alle priemgetallen te vinden wereldwijd in pamfletten werd gepubliceerd. Denk aan de (nota bene) gepensioneerd hoogleraar wiskunde die bij hoog en laag volhield een eenvoudig bewijs voor de vier-kleurenstelling te hebben gevonden.[9] Lach dus niet! In ieder van ons schuilt een crank.

Hoewel de overtuiging van een crank voor professionele wiskundigen en informatici meestal overduidelijk onwaar is, kunnen cranks soms opmerkelijk succesvol zijn in het overtuigen van leken. Een fameus voorbeeld is die keer dat in de Amerikaanse staat Indiana door intensief lobbyen van een crank in 1897 bijna een wet werd aangenomen (the Indiana Pi Bill) die verordonneerde hoe een cirkel kan worden gekwadrateerd, iets waarvan wiskundigen weten dat dit onmogelijk is. Een oplettende hoogleraar wiskunde, C.A. Waldo (1852-1926) wist het congres er nog net van te overtuigen dat de wet maar beter niet kon worden aangenomen.

Boeken vol zijn er geschreven over foute bewijzen, paradoxen, en cranks. Voor foute bewijzen verwijzen we naar [Barbeau 2000] en [Bunch 1997]. Voor cranks verwijzen we naar werk van Underwood Dudley, die, omdat hij brieven van cranks omwerkte tot publicaties (waarin correspondentie werden geanonimiseerd), nog verschillende keren voor de rechter is gesleept [Dudley 1992].

[6] 2018.
[7] 2018.
[8] 2018.
[9] De vier-kleurenstelling is waar, maar tot op heden komt het kortste bewijs neer op het checken van 1,936 grafen plus nog wat extra werk.

Deel I

Verzamelingenleer

Hoofdstuk 2

Naïeve verzamelingenleer

2.1 Terminologie

Wat verzamelingen zijn weet je eigenlijk al, want je begrijpt wat er bedoeld wordt wanneer iemand het heeft over de verzameling van alle Nederlanders boven 21 jaar, en je hebt een idee van wat een postzegelverzameling is.

In de wiskunde is een *verzameling* simpelweg een of andere collectie van dingen (Engels voor 'verzameling': *set*). De dingen die samen in een verzameling zitten heten de *leden* of *elementen* van die verzameling (Eng.: *members*, *elements*). We kunnen verzamelingen namen geven, bij voorbeeld door ze aan te duiden met hoofdletters van het Romeinse alfabet. We kunnen dan bij voorbeeld spreken over een of andere verzameling A.

Het begrip 'verzameling' is een grondbegrip dat zelf niet gedefinieerd kan worden. De omschrijving in termen van 'collectie van objecten' die we je zojuist hebben laten slikken is geen definitie maar een verschuiving van het definitie-probleem: immers, wat is een collectie? Net zo voor het begrip 'element': ook hier moeten we ons tevreden stellen met een vage kenschets. 'Verzameling' en 'element' zijn de meest fundamentele begrippen uit het wiskundige woordenboek: ze worden gebruikt om andere begrippen te definiëren, maar ze worden zelf niet gedefinieerd. Je word geacht ermee vertrouwd te zijn; als dat niet zo is raak je er vanzelf mee vertrouwd door dit hoofdstuk te lezen.

Verzamelingen spelen in wiskunde en logica een grote rol. Verzamelingenleer is een relatief jong onderdeel van de wiskunde, begonnen met het werk van Georg Cantor (1845–1918). Werken met verzamelingen maakt het mogelijk om snel complexe structuren te maken. Nieuwe objecten kunnen worden gecreëerd door lagere te verzamelen; die nieuwe objecten kunnen heel andere eigenschappen hebben dan hun elementen. Om dit laatste in te zien hoef je niets van wiskunde te weten: een Nederlandse postzegel kan de eigenschap hebben dat er een portret van Beatrix op staat, maar een verzameling van Nederlandse postzegels heeft die eigenschap niet.

Tot slot iets over de titel van het hoofdstuk. Waarom *naïeve* verzamelingenleer? Is er ook zoiets als "gepokt-en-gemazelde" of "door-de-wol-geverfde" verzamelingenleer? Ja, dat is er zeker. Om dat te begrijpen dien je eerst weten wat naïeve verzamelingenleer is en waarom we daar mee beginnen. Naïeve verzamelingenleer is de gewone huis-tuin-en-keuken ("vanilla") verzamelingenleer, waarbij de taal om verzamelingen te beschrijven en er over te redeneren, de zogenaamde *meta-taal*, gewone natuurlijke taal is. In naïeve verzamelingenleer worden bewijzen in gewone natuurlijke taal gegeven, en er worden geen beperkingen aan verzamelingconstructies opgelegd. Dat levert problemen op, want de inherente vaagheid (en soms ambiguïteit) van natuurlijke taal en het ontbreken van beperkingen aan verzamelingconstructies (het feit dat "alles maar moet kunnen") geeft aanleiding tot problemen en paradoxen (Sectie 4.8 op blz. 56). Om deze problemen en paradoxen het hoofd te bieden is er in het begin van de 20e eeuw gewerkt aan de zogenaamde *axiomatische verzamelingenleer*.

Axiomatische verzamelingenleer kan worden opgevat als een preciezere (logici zouden zeggen: meer formele) behandeling van de verzamelingenleer, waarbij de taal om verzamelingen te beschrijven, de zogenaamde *meta-taal*, predikaatlogica is. Bewijzen worden vanuit enkele onwrikbare principes, de zogenaamde *axioma's,* gegeven in de taal van de predikaatlogica, met behulp van het deductiesysteem (redeneermechanisme) van de predikaatlogica. Het ontstaan van problemen en paradoxen wordt vermeden door sommige producten van vermeende verzamelingconstructies te diskwalificeren als verzamelingen en ze *klassen* te noemen. Een merkwaardig gegeven is dat met deze klassen, die buiten het raamwerk van de axiomatische verzamelingenleer vallen, wel weer kan worden gewerkt als in de naïeve verzamelingenleer!! Ook om deze reden is de naïeve verzamelingenleer wel degelijk belangrijk.

2.2 Principes

De elementen van een verzameling A hoeven niets met elkaar gemeen te hebben (behalve dan het feit dat ze samen in A zitten). De elementen van een verzameling hoeven ook niet per se *materiële* dingen te zijn. Uit het feit dat de elementen van een verzameling abstracte dingen mogen zijn volgt dat die elementen bij voorbeeld zelf ook verzamelingen mogen zijn. Op deze mogelijkheid gaan we straks nader in.

Omdat de elementen van een verzameling niets met elkaar gemeen hoeven te hebben zouden we kunnen besluiten het getal 3, paus Johannes Paulus II en de stad Parijs samen in een verzameling te stoppen. Een manier om deze verzameling aan te duiden is:

$$A = \{3, \text{Johannes Paulus II}, \text{Parijs}\}.$$

De verzameling A wordt hier gekarakteriseerd door het opsommen van zijn elementen. Het feit dat het getal 3, de paus, en de stad Parijs samen in een verzameling zitten geven we aan met de accolades.

We kunnen verzamelingen op verschillende manieren definiëren. Wanneer het gaat om eindige verzamelingen (dat wil zeggen om verzamelingen waarvan je de leden kunt opsommen, zo dat die opsomming op een gegeven moment afgelopen is), dan kun je gewoon alle elementen *noemen*. Dit heet: definitie door opsomming. De introductie van de verzameling A hierboven is een voorbeeld van definitie door opsomming.

We kunnen verzamelingen ook invoeren door middel van *omschrijving*. Deze methode werkt ook voor oneindige verzamelingen. Voorbeeld:

$$B = \{x \mid x\ x \in \mathbb{N} \text{ en } x \text{ is even}\}.$$

B is de verzameling van alle natuurlijke getallen (getallen uit het rijtje 1, 2, 3, 4, enzovoorts) die deelbaar zijn door twee, ofwel: de verzameling van alle *even* natuurlijke getallen. Merk op dat B niet eindig is. B is verkregen door een deel te nemen van een andere verzameling, $\mathbb{N}$.

2.3 Notatie

We voeren nu de volgende notatie in. Voor elementen worden meestal kleine letters, en voor verzamelingen meestal hoofdletters gebruikt. (Als iemand dat nodig vindt, kan hij/zij van deze conventie afwijken.)

- $a \in B$ "a is een element van B."
- $a \notin B$ "a is geen element van B."
- $A \subseteq B$ "A is bevat in B"; "A is een deelverzameling van B" (dit wil zeggen: elk element van A is een element van B).
- $A \not\subseteq B$ "A is niet bevat in B"; "A is geen deelverzameling van B" (dit wil zeggen: niet elk element van A is een element van B).
- $A \subset B$ "A is echt bevat in B"; "A is een echte deelverzameling van B" (dit wil zeggen: elk element van A is een element van B en niet elk element van B is een element van A).
- $A \not\subset B$ "A is niet echt bevat in B"; "A is geen echte deelverzameling van B."

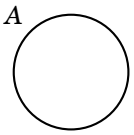
Deze notatie is internationaal gezien het meest gangbaar. In dit dictaat houden we ons hier aan.

In teksten worden de volgende notatie-varianten gebruikt:

- $\subset$ wordt gebruikt voor "is bevat in".
- $\subsetneq$ wordt gebruikt voor "is echt bevat in".

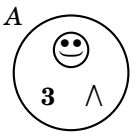
Wij gebruiken deze hier verder niet.

Om het denken over verzamelingen te vergemakkelijken is het nuttig om plaatjes te tekenen. Een verzameling A geven we als volgt aan met behulp van een cirkel:

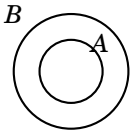


De punten die binnen de cirkel liggen zijn de elementen van A.

Wanneer A een eindige verzameling is, kunnen we de afzonderlijke elementen in de cirkel tekenen:



Het gegeven dat $A \subseteq B$ kan nu in het volgende plaatje worden uitgedrukt:



Als $A = B$ dan is het buitengebied leeg, als $A \subset B$ dan bevat het buitengebied een of meer elementen. We tekenen de plaatjes altijd zo algemeen mogelijk, dus we houden rekening met deze tweede mogelijkheid. Dit soort plaatjes heten Venn-diagrammen (naar de logicus J. Venn, 1834–1923). Plaatjes gelden echter nooit als bewijs. (Lees indien nodig nog eens de sectie "Plaatjes" op blz. 15.)

Verzamelingen die precies één element hebben worden *atomaire verzamelingen* of *singletons* genoemd (Eng.: *singletons*, *singleton sets*). Dus: wanneer a een of ander willekeurig ding is, dan is $\{a\}$ een atomaire verzameling. De verzameling $\{a\}$ wordt wel aangeduid als: "singleton a". Let op: er is verschil tussen a en $\{a\}$, tussen het ding a en de atomaire verzameling met a als element. Iets concreter: er is verschil tussen de paus en de atomaire verzameling met de paus als element. Johannes Paulus II is een persoon; de verzameling met Johannes Paulus II als enige element is geen persoon, maar een eigenschap, namelijk de eigenschap Johannes Paulus II te zijn. In een plaatje:

de paus singleton de paus

We hebben hierboven opgemerkt dat de elementen van verzamelingen zelf ook weer verzamelingen kunnen zijn. Dus: als a en b willekeurige dingen zijn, dan is $\{a, b\}$ een verzameling. Maar nu is $\{\{a, b\}\}$ ook een verzameling, namelijk: de verzameling met de verzameling $\{a, b\}$ als enige element. Let op: $\{\{a, b\}\}$ en $\{a, b\}$ zijn verschillende verzamelingen; $\{\{a, b\}\}$ is een atomaire verzameling (Eng.: singleton), $\{a, b\}$ is dat niet.

Opgave 2.1 Ga na of de volgende verzamelingen singletons zijn:

1. $\{\{a\}\}$
2. $\{\{a\},\{b\}\}$
3. $\{\{\{a\},\{b\}\}\}$.
4. $\{\{a\},\{a\}\}$
5. $\{\{a,b\},\{b,a\}\}$.

Opgave 2.2 Hier volgen wat beweringen over verzamelingen waarbij de zojuist ingevoerde notatie-afspraken worden gebruikt. Ga na wat de beweringen precies uitdrukken en of ze juist zijn. De objecten a en b zijn *verschillend*, en het zijn zelf geen verzamelingen.

1. $a \in \{a\}$
2. $a \subseteq \{a\}$
3. $a \subseteq a$
4. $\{a\} \subseteq \{a\}$
5. $\{a\} \subseteq \{\{a\}\}$
6. $a \in \{a,b\}$
7. $\{a\} \in \{a,b\}$
8. $\{a\} \subset \{a,b\}$
9. $\{a\} \subseteq \{a,b\}$
10. $\{a\} \subseteq \{a,\{a\}\}$.

2.4 Gelijkheid van verzamelingen

Dat twee verzamelingen A en B aan elkaar gelijk zijn, kunnen we als volgt uitdrukken:

$$A = B.$$

Dat twee verzamelingen A en B *niet* aan elkaar gelijk zijn drukken we als volgt uit:

$$A \neq B.$$

Als twee verzamelingen aan elkaar gelijk zijn dan hebben ze dezelfde elementen. Met andere woorden: als $A = B$ dan geldt voor iedere x: $x \in A$ dan en slechts dan als $x \in B$. Met andere woorden: als $A = B$ geldt voor iedere x: als $x \in A$ dan $x \in B$ *en* als $x \in B$ dan $x \in A$. Nog anders gezegd: als $A = B$ dan geldt $A \subseteq B$ en $B \subseteq A$.

In het vervolg zullen we 'dan en slechts dan als' afkorten als *desda* (Engels voor 'dan en slechts dan als': *if and only if*; voor de afkorting 'desda': *iff*). In plaats van 'desda' wordt ook wel het symbool $\Longleftrightarrow$ gebruikt. De bewering 'p dan en slechts dan als q' is een combinatie van 'p indien q' (dat wil zeggen: 'als q dan p'), soms genoteerd als '$p \Leftarrow q$', en 'p slechts indien q' (dat wil zeggen 'als p dan q'), met notatie '$p \Rightarrow q$'.

Voor de hand liggende vraag: geldt nu ook *omgekeerd*, dat als twee verzamelingen A en B dezelfde elementen hebben, A gelijk is aan B? In de verzamelingenleer geldt de afspraak dat dit inderdaad zo is. Deze afspraak is vervat in het zogenaamde *extensionaliteits-axioma* of *axioma van uitgebreidheid*.

Axioma 2.1 (Extensionaliteit)

$A = B$ desda A en B dezelfde elementen hebben.

Het extensionaliteits-axioma kan als volgt worden geparafraseerd: $A = B$ desda voor alle x geldt: $x \in A$ desda $x \in B$. Wat het axioma in feite zegt is dat de identiteit van een verzameling volledig bepaald is door zijn elementen. Twee verzamelingen zijn alleen verschillend wanneer er een ding valt aan te wijzen dat element is van de ene maar niet van de andere verzameling. Uit het extensionaliteitsaxioma volgt meteen:

- $\{a,b\} = \{b,a\}$
- $\{a,b,c\} = \{b,c,a\}$.

Het maakt kennelijk niet uit in welke volgorde je de elementen van een verzameling opsomt. De elementen van een verzameling zijn *ongeordend*. Verder volgt uit het extensionaliteitsaxioma:

- $\{a,a\} = \{a\}$.

Immers, er valt geen element aan te wijzen dat in $\{a,a\}$ zit en niet in $\{a\}$, en omgekeerd net zo. Dus: het heeft geen zin om bij het opsommen van een verzameling een bepaald element meerdere keren te noemen.

2.5 De lege verzameling

Het ligt niet in het begrip verzameling opgesloten dat een verzameling minstens één element heeft. We kunnen ook verzamelingen met nul elementen bekijken. Laten we een verzameling met nul elementen een lege verzameling noemen. Het extensionaliteits-axioma levert nu op dat er *precies één* lege verzameling is. Immers, stel dat er meer dan één lege verzameling zou zijn. Dan zou je twee lege verzamelingen A en B kunnen nemen die dan—volgens het extensionaliteits-axioma—van elkaar zouden moeten verschillen in het feit dat A een element heeft dat B niet heeft of omgekeerd. Maar dit kan nu juist niet, omdat A en B allebei leeg zijn. Dus: A en B moeten—in tegenspraak met wat we hadden aangenomen—aan elkaar gelijk zijn. De slotsom is dat er precies één lege verzameling is. We noemen deze verzameling *de lege verzameling*. De lege verzameling wordt vaak aangeduid als $\emptyset$. Soms wordt ook wel de notatie $\{\}$ gebruikt, maar wij houden het op $\emptyset$.

Bij de eerste kennismaking met de lege verzameling is het even wennen. De lege verzameling is een ding (zij het dan een abstract ding). Dit ding kan dus zelf voorkomen als element van verzamelingen. We kunnen dus bij voorbeeld hebben: $\{\emptyset\}$, ofwel: singleton de lege verzameling. Weer moeten we bedacht zijn op het verschil tussen $\emptyset$ en $\{\emptyset\}$. De lege verzameling is niet gelijk aan singleton de lege verzameling. Immers: $\emptyset$ heeft geen elementen, en $\{\emptyset\}$ heeft er precies één. We hebben:

- $\emptyset \in \{\emptyset\}$.

Verder heeft de lege verzameling de eigenschap dat hij bevat is in elke andere verzameling. Immers: omdat $\emptyset$ geen elementen heeft zal altijd gelden: als A een verzameling is, dan is elk element van $\emptyset$ element van A.

Opgave 2.3 1. Bewijs dat de lege verzameling uniek is. (Hint: gebruik het extensionaliteitsaxioma, blz. 23.)

2. Ga na dat $\emptyset$ *echt* bevat is in elke verzameling behalve $\emptyset$.

Opgave 2.4 Wat drukken de volgende beweringen uit? Welke ervan zijn juist en welke niet?

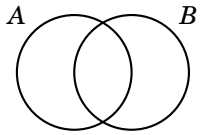
1. $\emptyset \subseteq \{a\}$
2. $\emptyset \subseteq \{\emptyset\}$
3. $\emptyset \in \{\emptyset\}$
4. $\emptyset \in \emptyset$
5. $\emptyset \notin \emptyset$
6. $\emptyset \in \{a\}$.

2.6 Vereniging, doorsnede en verschil

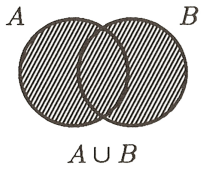
Laat A en B twee verzamelingen zijn. Dan is de *vereniging* (Eng.: *union* of *join*) van A en B als volgt gedefinieerd.

Definitie 2.1 *De* **vereniging** *van A en B* $\stackrel{\text{def}}{=}$ *de verzameling van alle elementen uit A plus alle elementen uit B.*

We noteren de vereniging van A en B als $A \cup B$. Het is nuttig om plaatjes (Venn-diagrammen) te tekenen. We tekenen een Venn-diagram altijd zo algemeen mogelijk:



Nu is $A \cup B$ het gearceerde gebied:



Een paar voorbeelden van het gebruik van $\cup$:

- $\{a,b\} \cup \{a\} = \{a,b\}$
- $\{a,b\} \cup \{a,c\} = \{a,b,c\}$
- $\{a\} \cup \{b\} = \{a,b\}$
- $\{a\} \cup \{\{a\}\} = \{a,\{a\}\}$
- $\{a,b\} \cup \emptyset = \{a,b\}$.

Het verenigen van twee verzamelingen is een *bewerking* die je op verzamelingen uitvoert, en waarvan het resultaat een nieuwe verzameling is. Zo'n bewerking wordt een *operatie* genoemd. De operatie 'verenigen' (aangeduid met het symbool $\cup$) heeft een aantal elementaire eigenschappen die we hier opsommen:

- $A \cup A = A$ (deze eigenschap heet *idempotentie*)
- $A \cup B = B \cup A$ (deze eigenschap heet *commutativiteit*)
- $A \cup (B \cup C) = (A \cup B) \cup C$ (deze eigenschap heet *associativiteit*).

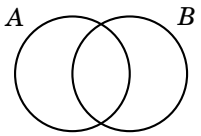
Het bewijs van de idempotentie van $\cup$: een willekeurig ding x zit in $A \cup A$ desda x in een van de twee verenigde verzamelingen zit, dus: desda x in A zit of in A, dat wil zeggen desda x in A zit. Dus $A \cup A = A$.

Opgave 2.5 Bewijs op dezelfde manier dat $\cup$ commutatief en associatief is.

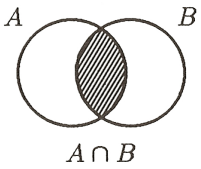
Uit de associativiteit van $\cup$ volgt dat we willekeurig veel verzamelingen kunnen verenigen zonder ons druk te maken om de haakjes (dat wil zeggen om de *volgorde* bij het toepassen van de verenigings-operatie). We kunnen dus in plaats van $A \cup (B \cup C)$ of $(A \cup B) \cup C$ zonder bezwaar schrijven: $A \cup B \cup C$.

Definitie 2.2 *De* **doorsnede** *of* **doorsnijding** *(Eng.:* intersection *of* meet*) van twee verzamelingen A en B* $\stackrel{\text{def}}{=}$ *de verzameling van de dingen die zowel element van A als van B zijn.*

De doorsnijding van A en B wordt aangegeven als: $A \cap B$. Weer is een plaatje nuttig. Laat dit de verzamelingen A en B zijn:



Dan is het gearceerde gebied de verzameling $A \cap B$:



Een paar voorbeelden (neem aan dat a, b en c onderling *verschillende* dingen zijn):

- $\{a\} \cap \{b\} = \emptyset$
- $\{a,b,c\} \cap \{a,b\} = \{a,b\}$
- $\{a,b\} \cap \{b,c\} = \{b\}$
- $\{a\} \cap \{a\} = \{a\}$
- $\{a,b,c\} \cap \emptyset = \emptyset$.

Opgave 2.6 1. Beschouw de jongste oliebollenbakker onder schaatsers en de jongste schaatser onder oliebollenbakkers. Kunnen dit twee verschillende mensen zijn?

2. Dezelfde vraag voor de beste oliebollenbakker onder schaatsers en de beste schaatser onder oliebollenbakkers.

Net als 'verenigen' heeft 'doorsnijden' de volgende eigenschappen:

- $A \cap A = A$ (idempotentie)

– $A \cap B = B \cap A$ (commutativiteit)

– $A \cap (B \cap C) = (A \cap B) \cap C$ (associativiteit).

Bewijs van commutativiteit: zij x willekeurig. Dan geldt het volgende: $x \in A \cap B$ desda $x \in A$ en $x \in B$. Omdat het woord "en" in de natuurlijke taal commutatief is (dit is de crux van het bewijs), mogen we nu schrijven $x \in B$ en $x \in A$, met andere woorden $x \in B \cap A$. Omdat x willekeurig was geldt $A \cap B = B \cap A$.

Het lijkt kinderachtig, maar veel bewijzen zijn voor een groot deel nou eenmaal verkapte invuloefeningen, toepassingen van definities, of het vertalen van formele constructies in de objecttaal (hier: $A \cap B$) naar informele constructies in de meta-taal (hier: $x \in A$ en $x \in B$).

Merk wel op dat de meta-expressie hier over individuen, x, gaat en de object-expressie niet. Bovenstaand bewijsje van commutativiteit is dus al geen invuloefening meer en er is al een zeker niveau van indirectie. In dit geval wordt de indirectie veroorzaakt door het feit dat het bewijs via individuen verloopt.

Opgave 2.7 Iemand geeft het volgende "bewijs" van de commutativiteit van doorsnede:

> $A \cap B = B \cap A$ want $A \cap B$ betekent "A doorsneden met B", wat hetzelfde is als "B doorsneden met A", wat hetzelfde is als $B \cap A$.

Waarom is dit geen goed bewijs?

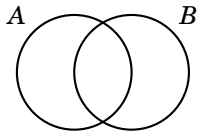
Opgave 2.8 Bewijs de idempotentie en associativiteit van $\cap$.

We kunnen nu ook een argument geven waarom een abstract object als $\emptyset$ handig is. Stel dat A en B verzamelingen zijn zonder gemeenschappelijke elementen. Ook in dit geval willen we toch dat $A \cap B$ een verzameling is: dat is nu dus $\emptyset$.

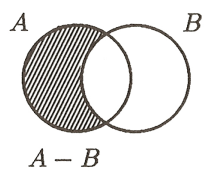
De derde elementaire operatie op verzamelingen die we behandelen is het nemen van het *verschil* van twee verzamelingen A en B (Eng.: *the difference of A and B*).

Definitie 2.3 *Het* **verschil** *van A en B* $\stackrel{\text{def}}{=}$ *de verzameling van alle elementen van A die niet in B zitten.*

Notatie voor het verschil van A en B: $A \setminus B$ (Angelsaksische en Russische literatuur), of $A - B$ (Europese literatuur). We tekenen weer een plaatje. Hier zijn weer de verzamelingen A en B:



$A - B$ is nu het gearceerde gedeelte:



Voorbeelden (neem weer aan dat a, b, en c onderling verschillende dingen zijn):

– $\{a\} - \{b\} = \{a\}$

– $\{a,b,c\} - \{a,b\} = \{c\}$

– $\{a,b\} - \emptyset = \{a,b\}$

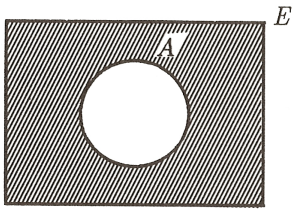
– $\{a,\{a\}\} - \{a\} = \{\{a\}\}$

– $\{a,\{a\}\} - \{\{a\}\} = \{a\}$

– $\emptyset - \{a,b,c\} = \emptyset$.

Opgave 2.9 Is de verschil-operatie idempotent? Commutatief? Associatief?

Wanneer A een deelverzameling is van E noemen we het verschil van E en A ook wel: het *complement* van A ten opzichte van E. Het gearceerde gedeelte in het plaatje is het complement van A ten opzichte van E:



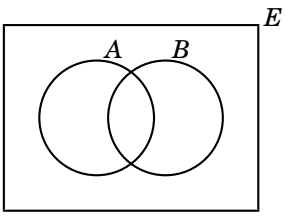
Wanneer duidelijk is ten opzichte van *welke* verzameling er complementen worden genomen kunnen we het vermelden van die grotere verzameling ook achterwege laten.

Definitie 2.4 (Universum) *Het* universum *is de verzameling ten opzichte waarvan unaire complementen worden genomen. Als het universum U is, dan definiëren we, voor elke $A \subseteq U$,*

$$A^c = U \setminus A$$

Andere notaties die ook wel worden gebruikt zijn: A' en $\overline{A}$.

Opgave 2.10 Beschouw het volgende plaatje:



Geef de volgende verzamelingen aan door middel van arceren (teken zo nodig nieuwe plaatjes):

1. A^c
2. $(A^c)^c$
3. $A^c \cup B^c$
4. $A^c \cap B^c$
5. $(A \cup B)^c$
6. $(A \cap B)^c$.

De complementen zijn steeds ten opzichte van E.

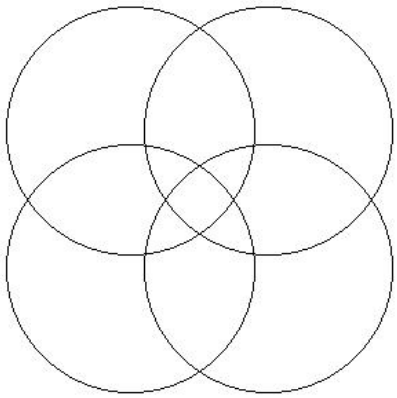
Opgave 2.11 Zij verzamelingen A, B en C willekeurig. Welke van de onderstaande gelijkheden zijn waar?

– Als de gelijkheid waar is, geef dan 2x een gearceerd Venn-diagram, één keer voor de linkerkant van de gelijkheid, en één keer voor de rechterkant van de gelijkheid. Je kunt ook één Venn-diagram tekenen en schuin links cq. schuin rechts arceren, of inkleuren met twee verschillende kleuren potlood.

– Geef anders een minimaal tegenvoorbeeld.

1. $(A \cap B) \cup C = (A \cup B) \cap (B \cup C)$.
2. $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$.
3. $(A \cup B) \setminus C = (A \setminus C) \cup B$.
4. $(A \cap B) \setminus C = (A \setminus C) \cap B$.
5. $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$.
6. $A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$.

Opgave 2.12 1. Waarom is onderstaand figuur geen representatief Venn-diagram voor vier verzamelingen?



2. Teken een representatief Venn-diagram voor vier verzamelingen.
3. Is het mogelijk een representatief Venn-diagram voor n verzamelingen te tekenen, waarbij n een willekeurig geheel getal is, groter dan 1?

Stelling 2.1 (Wetten van De Morgan) *Zij A, B, X willekeurig, en $A \subseteq X$, $B \subseteq X$. We nemen aan dat X het universum is. Er geldt:*

$$(A \cup B)^c = A^c \cap B^c \quad (1)$$
$$(A \cap B)^c = A^c \cup B^c \quad (2)$$

Bewijs: We bewijzen eerst (1). Daarvan bewijzen we eerst de inclusie van links naar rechts, i.e., we bewijzen eerst

$$(A \cup B)^c \subseteq A^c \cap B^c.$$

Zij $x \in (A \cup B)^c$ willekeurig.

– Per definitie van complement-teken (kleine c bovenaan), geldt dus $x \notin A \cup B$.
– Per definitie van het $\notin$-teken volgt dat x geen element van $A \cup B$ is.
– Per definitie van het verenigingsteken $\cup$ volgt dat $x \in A$ of $x \in B$.
– Per natuurlijke taal volgt nu dat niet $x \in A$ en niet $x \in B$.
– Per definitie van het $\notin$-teken volgt dat $x \notin A$ en $x \notin B$.
– Per definitie van complement-teken volgt weer $x \in A^c$ en $x \in B^c$.
– Per definitie van het doorsnijdingsteken, $\cap$, volgt tenslotte $x \in A^c \cap B^c$.

Omdat x willekeurig was, geldt $(A \cup B)^c \subseteq A^c \cap B^c$.

Nu bewijzen we de inclusie van rechts naar links, i.e., we bewijzen nu

$$A^c \cap B^c \subseteq (A \cup B)^c.$$

Om het bewijs wat sneller te laten verlopen staan de afzonderlijke stapjes nu niet meer onder elkaar maar zijn ze weergegeven als doorlopende tekst.

Zij $x \in A^c \cap B^c$ willekeurig. Dan $x \in A^c$ en $x \in B^c$. Dus $x \notin A$ en $x \notin B$. Dus niet $x \in A$ en niet $x \in B$. Dus niet: $x \in A$ of $x \in B$. Dus niet $x \in A \cup B$. Dus $x \notin A \cup B$. Dus $x \in (A \cup B)^c$. Omdat x willekeurig was, geldt $A^c \cap B^c \subseteq (A \cup B)^c$.

Samen geven deze twee inclusies

$$(A \cup B)^c = A^c \cap B^c,$$

i.e, (1). Het bewijs van (2) verloopt analoog en laten we als opgave. □

Opgave 2.13 Bewijs Stelling 2.1, bewering (2).

In dit geval gelden de implicaties twee kanten op en hadden we het bewijs van DeMorgan (1) dus sneller kunnen geven door meteen die equivalenties te volgen. Dus voor willekeurige $x \in X$ zou dan gelden

	$x \in (A \cup B)^c$	
$\Leftrightarrow$	$x \notin A \cup B$	[per def. complement]
$\Leftrightarrow$	niet : $x \in A \cup B$	[per def. "$\notin$"]
$\Leftrightarrow$	niet : $x \in A$ of $x \in B$	[per def. "$\cup$"]
$\Leftrightarrow$	niet $x \in A$ en niet $x \in B$	[per natuurlijke taal]
$\Leftrightarrow$	$x \notin A$ en $x \notin B$	[per def. "$\notin$"]
$\Leftrightarrow$	$x \in A^c$ en $x \in B^c$	[per def. complement]
$\Leftrightarrow$	$x \in A^c \cap B^c$	[per def. "$\cap$"]

Echter, dit kun je alleen doen als je ook echt heel zeker weet dat elke equivalentie die je opschrijft ook echt waar is, i.e, geldig is in beide richtingen.

Hier is een voorbeeld waarbij het klakkeloos opschrijven van equivalenties leidt tot fouten.

Voorbeeld 2.1 (Misbruik van equivalentie) Iemand schrijft op:

$$y > 0 \text{ en } \sqrt{y} = x \quad \Leftrightarrow \quad y > 0 \text{ en } y = x^2,$$

terwijl in feite alleen geldt

$$y > 0 \text{ en } \sqrt{y} = x \;\Rightarrow\; y > 0 \text{ en } y = x^2.$$

In dit voorbeeld is duidelijk waar het misloopt (kwadraten hebben twee oplossingen) maar als bewijzen ingewikkelder worden zie je dit soort dingen al snel over het hoofd, omdat per bewijsstap steeds de geldigheid van twee implicaties tegelijkertijd geverifieerd moet worden.

Opgave 2.14 Bewijs Stelling 2.1, bewering (2).

Opgave 2.15 Bewijs de geldige beweringen in Opgave 2.11 op de manier zoals in het bewijs van de Stelling van De Morgan.

Voorbeeld 2.2 Bewijs: $A \subseteq B$ als en alleen als $A \cup B = B$.

Uitwerking: neem eerst aan dat $A \subseteq B$ geldt. Te bewijzen dat $A \cup B = B$. Om deze gelijkheid te aan te tonen bewijzen we weer twee inclusies, te weten $A \cup B \subseteq B$ en $B \subseteq A \cup B$.

We bewijzen allereerst de inclusie $A \cup B \subseteq B$. Stel $x \in A \cup B$ willekeurig. Dan $x \in A$ of $x \in B$. Als $x \in B$ dan zijn we klaar. Als $x \in A$, gebruiken we de aanname $A \subseteq B$, om te concluderen dat in dit geval ook $x \in B$. Omdat x willekeurig was mogen we nu concluderen dat $A \cup B \subseteq B$.

Als tweede bewijzen we de omgekeerde inclusie, $B \subseteq A \cup B$. Stel $x \in B$ willekeurig. Dan zeker $x \in A$ of $x \in B$. Dus $x \in A \cup B$. Omdat x willekeurig was mogen we nu concluderen dat $B \subseteq A \cup B$. De twee inclusies geven samen $A \cup B = B$.

Nu het omgekeerde te bewijzen, namelijk, dat $A \subseteq B$ volgt uit $A \cup B = B$. Neem daarvoor $x \in A$ willekeurig. Dus zeker $x \in A$ of $x \in B$. Dus $x \in A \cup B$. Vanwege onze aanname $A \cup B = B$ mogen we nu concluderen dat $x \in B$. Omdat x willekeurig was, volgt nu $A \subseteq B$. □

Opgave 2.16 Zij A, B, X willekeurige verzamelingen, en $A \subseteq X$, $B \subseteq X$. We nemen aan dat X het universum is. Bewijs de volgende beweringen op de manier zoals in het bewijs van de Stelling van De Morgan.

1. $A \subseteq B$ als en alleen als $A \cup B = B$.
2. $A \subseteq B$ als en alleen als $A \cap B = A$.
3. $A \subseteq B^c$ als en alleen als $A \cap B = \emptyset$.
4. $A^c \subseteq B$ als en alleen als $A \cup B = X$.
5. $A \subseteq B$ als en alleen als $B^c \subseteq A^c$.
6. $A \subseteq B^c$ als en alleen als $B \subseteq A^c$.

Opgave 2.17 Zij A, B, C, X willekeurige verzamelingen, en $A \subseteq B \subseteq C \subseteq X$. We nemen aan dat X het universum is. Bewijs zoals boven:

a) $B \setminus A \subseteq C \setminus A$

b) $C \setminus (B \setminus A) = A \cup (C \setminus B)$

Opgave 2.18 Bestaan er verzamelingen A, B en C, zó dat $A \cap B \neq \emptyset$, $A \cap C = \emptyset$ en $(A \cap B) \setminus C = \emptyset$?

Opgave 2.19 Zij $A_1 \subseteq \cdots \subseteq A_n$ verzamelingen, $n \geq 2$, zó dat $A_n \subseteq A_1$. Bewijs $A_1 = A_n$.

2.7 Vereniging en doorsnijding van collecties

We generaliseren daarom de operaties *verenigen* en *doorsnijden*. Stel bijvoorbeeld dat we de volgende oneindige verzameling van verzamelingen hebben:

$$C = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, \ldots\}. \tag{3}$$

Elk van de A_i kan zelf zowel eindig als oneindig zijn. Een verzameling van verzamelingen wordt vaak een *collectie* genoemd.. De vereniging van een collectie wordt als volgt gedefinieerd:

Definitie 2.5 (Vereniging van een collectie)

$$\bigcup C =_{Def} \{x \mid x \textit{ is element van } \underline{\textit{tenminste één}}\ A \in C\}.$$

Met deze constructie is het ook mogelijk een collectie te verenigen die bestaat uit overaftelbaar[1] veel verzamelingen:

$$\bigcup \{\{x\} \mid x \in \mathbb{R}\} = \mathbb{R}.$$

is een flauw voorbeeld, maar laat zien dat het kan.

Als C aftelbaar is, zoals in (3), dan wordt $\bigcup C$ meestal genoteerd als

$$A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5 \cup \ldots \quad \text{of als} \quad \bigcup_{n=1}^{\infty} A_i.$$

Soms kom je ook wel eens de volgende notatie tegen:

$$\bigcup_{i \in I} A_i,$$

waarbij I elke set kan zijn. Dit is hetzelfde als $\bigcup \{A_i \mid i \in I\}$. Bijvoorbeeld:

$$\bigcup_{x \in \mathbb{R}} \{\{x\}\} = \mathbb{R}.$$

Geheel analoog wordt de doorsnede van C gedefinieerd:

Definitie 2.6 (Doorsnede van een collectie)

$$\bigcap C =_{Def} \{x \mid x \textit{ is element van } \underline{\textit{alle}}\ A \in C\}.$$

Opgave 2.20 1. Schrijf $A \cup B$ als de vereniging van een collectie verzamelingen.

[1] Het begrip *overaftelbaar* wordt pas ingevoerd in het volgende hoofdstuk. Het is echter onhandig vereniging en doorsnijding van aftelbare en overaftelbare collecties te behandelen ná beschouwingen over aftelbaarheid en overaftelbaarheid. De ervaring leert namelijk dat je niet hoeft te weten hoe groot een collectie is om een vereniging of doorsnede te kunnen bepalen.

2. Laat $(0,z)$ bestaan uit alle reële getallen tussen de 0 en de z. Bepaal

$$\bigcup\{(0,z) \mid z \in \mathbb{R}\},$$

en

$$\bigcap\{(0,z) \mid z \in \mathbb{R}\}.$$

Opgave 2.21 Laat $C = \{A_i\}_{i\in I}$ en $D = \{B_i\}_{i\in I}$ twee collecties zijn die geïndiceerd worden door dezelfde verzameling I. Laat $j \in I$ willekeurig. Bewijs:

1. Er geldt $A_j \subseteq \bigcup_{i\in I} A_i$ en $\bigcap_{i\in I} A_i \subseteq A_j$.
2. – $\bigcup_{i\in I}(A_i \cup B_i) = (\bigcup_{i\in I} A_i) \cup (\bigcup_{i\in I} B_i)$.
   – $\bigcap_{i\in I}(A_i \cap B_i) = (\bigcap_{i\in I} A_i) \cap (\bigcap_{i\in I} B_i)$.
3. Als voor alle $i \in I$ geldt dat $A_i \subseteq B_i$, dan ook

$$\bigcup_{i\in I} A_i \subseteq \bigcup_{i\in I} B_i \quad \text{en} \quad \bigcap_{i\in I} A_i \subseteq \bigcap_{i\in I} B_i.$$

4. Laat X een verzameling in het verzamelingstheoretische universum zijn. Dan
   – $\bigcup_{i\in I}(A_i \cap X) = (\bigcup_{i\in I} A_i) \cap X$.
   – $\bigcap_{i\in I}(A_i \cup X) = (\bigcap_{i\in I} A_i) \cup X$.
5. Wetten van DeMorgan.
   – $X \setminus \bigcap_{i\in I} A_i = \bigcup_{i\in I}(X \setminus A_i)$.
   – $X \setminus \bigcup_{i\in I} A_i = \bigcap_{i\in I}(X \setminus A_i)$.

Opgave 2.22 Bepaal:

1. $\bigcup \emptyset$
2. $\bigcup\{\emptyset\}$
3. $\bigcup\{\emptyset,\{\emptyset\}\}$
4. $\bigcup\{\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}$
5. $\bigcap\{\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}$
6. $\bigcup\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}$.

2.8 Deel- en machtsverzamelingen

De notatie $A \subseteq B$ voor "A is een deelverzameling van B" hebben we hierboven al ingevoerd. "A is een deelverzameling van B" betekent per definitie: elk element van A is een element van B.

Nu is de verzameling van alle deelverzamelingen van een verzameling A zelf weer een verzameling. Deze verzameling noemen we de *machtsverzameling* van A (Eng.: *the power set of A*). Notatie: 2^A, of ook wel: $\mathscr{P}(A)$. De eerste notatie is eigenlijk de meest consequente, omdat de notatie Y^X al beschikbaar is. De verzameling Y^X staat per definitie voor alle functies van X naar Y. In dit geval: alle functies van A naar 2. In de verzamelingenleer is 2 gedefinieerd als $\{0,1\}$.

De machtsverzameling van een verzameling A is als volgt gedefinieerd:

Definitie 2.7 $2^A \stackrel{\text{def}}{=} \{B \mid B \subseteq A\}$.

Merk op dat voor elke verzameling A geldt dat de lege verzameling element is van 2^A. Immers: $\emptyset$ is een deelverzameling van elke verzameling, dus ook van A. Verder geldt voor elke verzameling A dat A een element is van 2^A. De verzameling A zelf is immers ook een deelverzameling van A.

Voorbeelden van machtsverzamelingen (we nemen weer aan dat a, b en c onderling verschillende dingen zijn):

– $2^\emptyset = \{\emptyset\}$
– $2^{\{a\}} = \{\emptyset,\{a\}\}$
– $2^{\{a,b\}} = \{\{a,b\},\{a\},\{b\},\emptyset\}$
– $2^{\{a,b,c\}} = \{\{a,b,c\},\{a,b\},\{a,c\},\{b,c\},\{a\},\{b\},\{c\},\emptyset\}$.

Uit de voorbeelden lezen we af: de machtsverzameling van een verzameling zonder elementen heeft 1 element ; die van een verzameling met 1 element heeft 2 elementen; die van een verzameling met 2 elementen heeft 4 elementen; die van een verzameling met 3 elementen heeft 8 elementen. Algemeen (voor eindige verzamelingen): als een verzameling A n elementen heeft dan heeft 2^A 2^n elementen. Het bewijs zullen we hier nog niet geven (zie Opgave 7.7).

Opgave 2.23 Bepaal of de volgende beweringen correct zijn. Geef als antwoord een bewijs of tegenvoorbeeld. Zij A willekeurig.

a) $A \in 2^A$.

b) $A \subseteq 2^A$.

c) $\{A\} \in 2^A$.

d) $\{A\} \subseteq 2^A$.

e) $\emptyset \in 2^A$.

f) $\emptyset \subseteq 2^A$.

g) De verzameling $2^\emptyset$ is leeg.

h) De verzameling $\{\emptyset,\{\emptyset\}\}$ bevat twee verschillende elementen.

Opgave 2.24 Bewijs dat als $A \subseteq B$, dan $2^A \subseteq 2^B$.

Opgave 2.25 Bewijs hetzelfde nog een keer, maar dan door een machtsverzameling te zien als een functie van de oorspronkelijke verzameling naar de verzameling 2. (Herinner: $2 = \{0,1\}$.)

Opgave 2.26 Bewijs het omgekeerde van wat je in opdracht 2.24 hebt bewezen: als $2^A \subseteq 2^B$ dan $A \subseteq B$.

We kunnen de operatie van het nemen van de machtsverzameling van een verzameling *herhaald* toepassen. We kunnen dus spreken over 2^{2^A}, enzovoorts.

Voorbeeld (schrikt vooral niet; het is gewoon een kwestie van domweg de definitie toepassen):

$$2^{2^{\{\emptyset\}}} = 2^{\{\emptyset,\{\emptyset\}\}} = \{\{\emptyset,\{\emptyset\}\},\{\emptyset\},\{\{\emptyset\}\},\emptyset\}.$$

Opgave 2.27 Schrijf de volgende verzamelingen uit:

1. $2^{(2^{\emptyset})}$.
2. $2^{(2^{\{a\}})}$.
3. $2^{(2^{[2^{\{a\}}]})}$.
4. $2^{(2^{\{a,b\}})}$.

2.9 Geordende paren, rijtjes en producten

Wanneer we twee dingen a en b samenvoegen in een verzameling $\{a,b\}$ dan zijn—zoals we hebben gezien—die twee elementen ten opzichte van elkaar *ongeordend*. Dit wil zeggen: $\{b,a\}$ is dezelfde verzameling als $\{a,b\}$.

Het kan echter voorkomen dat we juist wel geïnteresseerd zijn in de volgorde van a en b. Neem het geval waarin a en b de coördinaten zijn voor een plaatsbepaling op het aardoppervlak, waarbij de eerste coördinaat de lengtegraad aangeeft, en de tweede de breedtegraad. a en b mogen nu niet worden verwisseld, want 15° Oosterlengte, 80° Noorderbreedte geeft een andere plaats aan dan 80° Oosterlengte, 15° Noorderbreedte. Een ander voorbeeld. Neem aan dat a en b schrijftekens zijn waarvan we de links-rechts volgorde willen vastleggen. Weer mogen a en b niet worden verwisseld.

Wanneer we de twee elementen a en b willen samenvoegen tot een paar waarvan de volgorde van belang is, dan spreken we van *het geordend paar* van a en b. De notatie die we invoeren is: $\langle a,b\rangle$. Ronde haken worden ook wel gebruikt; het geordend paar van a en b wordt dan aangegeven als: (a,b). Engels voor 'geordend paar' : *ordered pair*. Als a en b twee verschillende dingen zijn, dan hebben we: $\langle a,b\rangle \neq \langle b,a\rangle$. Verder geldt:

$$\langle a,b\rangle = \langle c,d\rangle \iff a = c \text{ en } b = d.$$

Als we een willekeurig geordend paar hebben kunnen we spreken over het *eerste lid* van het paar en het *tweede lid* van het paar. Het eerste lid van $\langle a,b\rangle$ is a, het tweede lid is b.

We kunnen het begrip 'geordend paar' generaliseren tot 'geordend n-tal' (Eng.: *ordered n-tuple*). Een geordend 1-tal $\langle a\rangle$ is niets anders dan het object a. En geordend 2-tal hebben we al gedefinieerd. Een geordend 3-tal is een rijtje $\langle a,b,c\rangle$. Voor geordende drietallen geldt:

$$\langle a,b,c\rangle = \langle e,f,g\rangle \iff a = e \text{ en } b = f \text{ en } c = g.$$

Voor langere rijtjes gaat het net zo.

We zien nu dat $\langle a\rangle \neq \langle a,a\rangle$. Immers: $\langle a\rangle$ is een geordend 1-tal en $\langle a,a\rangle$ een geordend paar. Dus: het meerdere keren opnemen van een en hetzelfde element maakt bij geordende rijtjes wel degelijk verschil.

Met behulp van het begrip 'geordend paar' kunnen we nu een nieuw begrip invoeren. Het *(Cartesisch) product* van twee verzamelingen A en B, notatie $A \times B$, is per definitie de verzameling van alle geordende paren waarvan het eerste element in A zit en het tweede element in B. Anders gezegd:

Definitie 2.8 $A \times B \stackrel{\text{def}}{=} \{\langle a,b\rangle \mid a \in A \textit{ en } b \in B\}$.

Het epitheton 'Cartesisch' is een hommage aan de Franse wiskundige en filosoof René Descartes (Cartesius) (1596–1650), die geordende paren gebruikte om punten in een vlak aan te duiden, in de door hem ontwikkelde analytische meetkunde.

Voorbeelden van Cartesische producten (laat a en b twee verschillende dingen zijn):

- $\{a\} \times \{a,b\} = \{\langle a,a\rangle,\langle a,b\rangle\}$
- $\{a,b\} \times \{a,b\} = \{\langle a,a\rangle,\langle b,b\rangle,\langle a,b\rangle,\langle b,a\rangle\}$
- $\{a,b\} \times \{a\} = \{\langle a,a\rangle,\langle b,a\rangle\}$.

Het is duidelijk dat als A en B verschillend zijn, $A \times B \neq B \times A$.

We kunnen ook Cartesische producten nemen van meer dan twee verzamelingen: dit levert verzamelingen geordende 3-tallen, 4-tallen, enzovoorts. Dus:

$$A \times B \times C = \{\langle a,b,c\rangle \mid a \in A, b \in B, c \in C\}.$$

Meestal wordt voor $A \times A$ een speciale notatie gebruikt: A^2. Net zo voor $A \times A \times A$: A^3, enzovoorts. In het algemeen staat A^n dus voor de verzameling van alle geordende n-tallen uit A.

Voorbeelden (a en b zijn weer twee verschillende dingen):

- $\{a,b\}^2 = \{\langle a,a\rangle,\langle b,b\rangle,\langle a,b\rangle,\langle b,a\rangle\}$
- $\{a,b\}^3 =$
  $\{\langle a,a,a\rangle,\langle a,a,b\rangle,\langle a,b,b\rangle,\langle b,b,b\rangle,\langle b,b,a\rangle,$
  $\langle b,a,a\rangle,\langle a,b,a\rangle,\langle b,a,b\rangle\}$.

Opgave 2.28 Hoeveel elementen bezit $\{1,2,3\}^3$?

Hoofdstuk 3

Relaties en functies

3.1 Relaties en hun eigenschappen

Wat relaties zijn weten we allemaal. We weten bij voorbeeld dat 'beminnen' een relatie is die tussen mensen kan bestaan of juist niet bestaan, dat 'eigenaar zijn van' een relatie is die tussen mensen en dingen kan bestaan, en dat 'groter zijn dan' een relatie is die tussen getallen kan bestaan.

Dit zijn allemaal voorbeelden van relaties tussen twee partners. We noemen ze daarom *twee-plaatsige relaties*. Er bestaan ook relaties die drie partners van node hebben. Een voorbeeld is 'geven aan': deze relatie geldt tussen een subject (de gever), een direct object (het gegevene) en een indirect object (de begunstigde). 'Geven aan' is een voorbeeld van—je vermoedde het al—een *drieplaatsige relatie*.

In het algemeen kun je de plaatsen die de partners in een relatie innemen niet straffeloos verwisselen. Uit het feit dat Jan Marietje bemint hoeft immers nog niet te volgen dat Marietje ook Jan bemint. Hieruit zien we dat tweeplaatsige relaties gelden tussen geordende paren, drieplaatsige relaties tussen geordende drietallen, enzovoorts.

Het inzicht dat we de partners in een relatie moeten beschouwen als de leden van een rijtje leidt tot een mooie abstracte kijk op relaties. Daarbij beschouwen we een relatie als een verzameling, en wel—je zag het al aankomen—als een verzameling van rijtjes. We definiëren de zaak als volgt.

Definitie 3.1 *R is een* **tweeplaatsige relatie** *tussen de elementen van A en de elementen van B* $\stackrel{\text{def}}{\iff}$ *R is een deelverzameling van het Cartesisch product van A en B.*

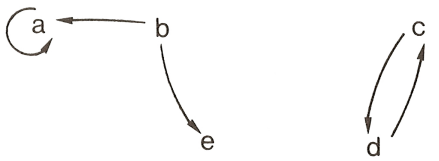
Anders gezegd: $R \subseteq A \times B$. We zeggen ook wel: R is een relatie *binnen* $A \times B$. Net zo voor een drieplaatsige relatie tussen elementen van A, B en C, enzovoorts.

Voorbeeld 3.1 Laat M een verzameling mensen zijn. Dan is de relatie 'beminnen' de verzameling B van geordende paren die bevat is in M^2 en die zodanig is dat $\langle a,b\rangle \in B$ desda persoon a persoon b bemint.

Neem voor het gemak even aan dat M bestaat uit 5 individuen, laten we zeggen: $M = \{a,b,c,d,e\}$. Stel dat a zichzelf bemint, en verder bemint b zowel a als e, en beminnen c en d elkaar. Neem aan dat er verder niet bemind wordt. Dan is de relatie B gelijk aan de volgende verzameling:

$$\{\langle a,a\rangle, \langle b,a\rangle, \langle b,e\rangle, \langle c,d\rangle, \langle d,c\rangle\}.$$

Als altijd is een plaatje nuttig. Wanneer R een relatie is binnen A^2, dan kunnen we elementen van R aangeven door het intekenen van pijlen in een plaatje van de verzameling A. Een pijl $\longrightarrow$ van x naar y betekent dat $\langle x,y\rangle$ element is van R. Voor ons voorbeeld levert dit het volgende plaatje op:



Hier komen nog een paar definities.

Definitie 3.2 *Als R een tweeplaatsige relatie is binnen $A \times B$, dan is het* **domein** *van R de verzameling van dingen uit A die tot zeker ding in B in relatie R staan.*

Gangbare notatie voor het domein (Eng.: *domain*) van R: $\mathrm{dom}(R)$.

Definitie 3.3 *Het* **bereik** *van R is de verzameling van dingen uit B waartoe zeker ding in A in de relatie R staat.*

Gangbare notatie voor het bereik (Eng.: *range*) van R: $\mathrm{rng}[R]$. We kunnen één en ander nu ook als volgt formuleren. Als $R \subseteq A \times B$, dan:

- $\mathrm{dom}(R) = \{a \mid a \in A$ en er is een $b \in B$ met $\langle a,b\rangle \in R\}$
- $\mathrm{rng}[R] = \{b \mid b \in B$ en er is een $a \in A$ met $\langle a,b\rangle \in R\}$.

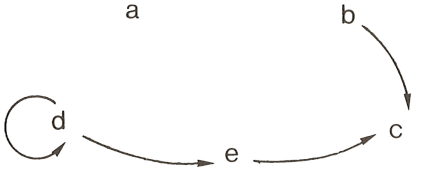
In woorden: $\mathrm{dom}(R)$ is de verzameling *eerste* leden van R, $\mathrm{rng}[R]$ de verzameling *tweede* leden. In het 'beminnen'-voorbeeld hierboven, waar B de relatie

$$\{\langle a,a\rangle, \langle b,a\rangle, \langle b,e\rangle, \langle c,d\rangle, \langle d,c\rangle\}$$

op $M = \{a,b,c,d,e\}$ is, hebben we:

- $\mathrm{dom}(B) = \{a,b,c,d\}$
- $\mathrm{rng}[B] = \{a,c,d,e\}$.

Opgave 3.1 Laat R de relatie zijn die met behulp van pijlen is getekend in het volgende plaatje. Wat zijn $\mathrm{dom}(R)$ en $\mathrm{rng}[R]$?



We kunnen nu *eigenschappen* van relaties gaan bestuderen. Dit onderwerp zullen we hier niet uitputtend behandelen; we noemen alleen een paar belangrijke eigenschappen van tweeplaatsige relaties waarvan zowel het domein als het bereik bevat is in een of andere verzameling A. Zulke relaties heten: tweeplaatsige relaties *op* A. Een tweeplaatsige relatie op A is dus een deelverzameling van A^2, ofwel: een relatie *binnen* A^2.

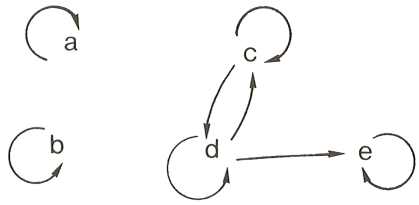
Opgave 3.2 Ga na dat 'groter dan' een tweeplaatsige relatie is op $\mathbb{N}$, waarbij $\mathbb{N}$ de verzameling van de natuurlijke getallen aanduidt, dat wil zeggen $\mathbb{N} = \{1,2,3,\ldots\}$.

Definitie 3.4 *Een relatie $R \subseteq A^2$ heet* **reflexief** *wanneer voor elke $a \in A$ geldt: $\langle a,a\rangle \in R$.*

Voorbeeld 3.2 De relatie 'waardering hebben voor' tussen de leden van een groep mensen is reflexief wanneer het zo is dat iedereen in die groep zichzelf waardeert (en mogelijkerwijs ook nog anderen).

Voorbeeld 3.3 De relatie 'groter of gelijk zijn aan' tussen natuurlijke getallen is reflexief: elk getal is immers groter dan of gelijk aan zichzelf.

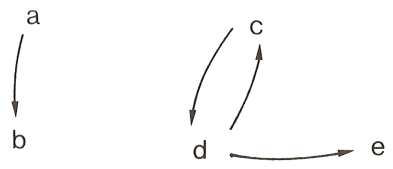
Voorbeeld 3.4 De relatie die met behulp van pijlen is aangegeven in het volgende plaatje is reflexief:



Definitie 3.5 *Een relatie $R \subseteq A^2$ heet* **irreflexief** *wanneer voor geen enkele $a \in A$ geldt dat $\langle a,a\rangle \in R$.*

Voorbeeld 3.5 De relatie 'groter zijn dan' tussen natuurlijke getallen is irreflexief: geen enkel getal is immers groter dan zichzelf.

Voorbeeld 3.6 De relatie die met behulp van pijlen is getekend in het volgende plaatje is irreflexief:



Opgave 3.3 Teken een plaatje van een relatie die noch reflexief, noch irreflexief is.

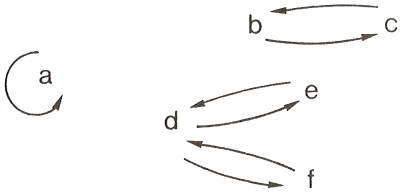
Definitie 3.6 *Een relatie $R \subseteq A^2$ is* **symmetrisch** *wanneer voor elk paar $\langle a,b\rangle \in R$ geldt: $\langle b,a\rangle \in R$.*

Voorbeeld 3.7 De relatie 'familie zijn van' tussen mensen is symmetrisch.

Voorbeeld 3.8 De relatie 'even oud zijn als' tussen mensen is symmetrisch.

Voorbeeld 3.9 De relatie 'beminnen' is—helaas—vaak *niet* symmetrisch.

Voorbeeld 3.10 In de situatie van het volgende plaatje is de relatie aangegeven door de pijlen symmetrisch:



Definitie 3.7 *Een relatie $R \subseteq A^2$ heet* **asymmetrisch** *wanneer voor geen enkel paar $\langle a,b\rangle \in R$ geldt dat $\langle b,a\rangle \in R$.*

Voorbeeld 3.11 De relatie 'ouder zijn dan' tussen mensen is asymmetrisch.

Voorbeeld 3.12 De relatie 'ouder zijn van' tussen mensen is asymmetrisch.

Opgave 3.4 Laat zien dat uit de definitie van *asymmetrie* volgt dat elke asymmetrische relatie irreflexief is.

Let op: het niet symmetrisch zijn van een relatie is iets anders dan het asymmetrisch zijn van een relatie. Zie de volgende opdracht.

Opgave 3.5 Teken een plaatje van een relatie die noch symmetrisch, noch asymmetrisch is.

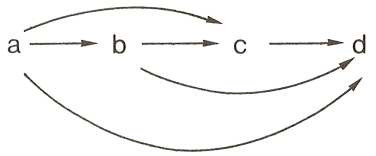
Definitie 3.8 *Een relatie $R \subseteq A^2$ heet* **transitief** *wanneer uit het gegeven dat $\langle a,b\rangle \in R$ en $\langle b,c\rangle \in R$ volgt: $\langle a,c\rangle \in R$.*

Voorbeeld 3.13 De relatie 'ouder zijn dan' tussen mensen is transitief: Als Jan ouder is dan Willem, en Willem is ouder dan Kees, dan is Jan ouder dan Kees.

Voorbeeld 3.14 De relatie 'groter zijn dan' tussen natuurlijke getallen is transitief.

Voorbeeld 3.15 De relatie 'groter dan of gelijk zijn aan' tussen natuurlijke getallen is transitief.

Voorbeeld 3.16 De relatie die met behulp van pijlen is getekend in het volgende plaatje is transitief:



Definitie 3.9 *Een relatie $R \subseteq A^2$ heet* **intransitief** *wanneer voor geen enkele a, b en c in A geldt: $\langle a,b\rangle \in R$, $\langle b,c\rangle \in R$ en $\langle a,c\rangle \in R$.*

Voorbeeld 3.17 De relatie 'vader zijn van' tussen mensen is intransitief.

Weer geldt: er bestaan relaties die noch transitief, noch intransitief zijn.

Opgave 3.6 Teken een plaatje van een relatie die noch transitief, noch intransitief is.

Definitie 3.10 *Een relatie $R \subseteq A^2$ heet* **antisymmetrisch** *wanneer voor elke a, b in A geldt: als $\langle a,b\rangle \in R$ en $\langle b,a\rangle \in R$, dan $a = b$.*

Voorbeeld 3.18 De relatie $\leq$ op de verzameling $\mathbb{N}$ van de natuurlijke getallen is antisymmetrisch.

Opgave 3.7

1. Laat zien dat een asymmetrische relatie altijd anti-symmetrisch is.

2. Laat zien dat het omgekeerde niet geldt. Je kunt dit laten zien door een voorbeeld te geven van een anti-symmetrische relatie die niet asymmetrisch is.

Opgave 3.8 Beschouw een willekeurige verzameling A. Merk op dat $\subseteq$ een tweeplaatsige relatie is op 2^A. Maak aan de hand van de hierboven gegeven definities een lijstje van de eigenschappen van deze relatie. Ook $\subset$ is een tweeplaatsige relatie op 2^A. Maak ook voor deze relatie een eigenschappenlijstje.

Zoals de laatste opdracht illustreert kunnen relaties allerlei combinaties vertonen van eigenschappen.

Definitie 3.11 *Een relatie die transitief, reflexief en antisymmetrisch is wordt een* **partiële orde** *genoemd.*

Definitie 3.12 *Een relatie die transitief, irreflexief en asymmetrisch is heet een* **strikte partiële orde**.

In feite is elke relatie die transitief en irreflexief is tevens asymmetrisch (ga dit na). Net zo is elke relatie die transitief en asymmetrisch is tevens irreflexief (ga na). Hieruit volgt dat het in de definitie van *strikte partiële orde* voldoende is om naast transitiviteit ofwel irreflexiviteit ofwel asymmetrie te eisen.

Opgave 3.9 Laat zien dat de relatie $\leq$ op de verzameling $\mathbb{N}$ van de natuurlijke getallen een partiële orde is.

Opgave 3.10 Laat zien dat elke strikte partiële orde bevat is in een partiële orde.

3.2 Functies

Een functie is een manier om elk van de elementen van een bepaalde verzameling in verband te brengen met een element van een andere verzameling. Voorbeelden van functies ken je uit de krant; de grafiekjes waarin de werkloosheidscijfers vanaf een jaar geleden tot nu maand voor maand zijn af te lezen zijn functies. Elke maand wordt in verband gebracht met een getal rond de 800.000 dat het aantal geregistreerde werklozen in die maand aangeeft.

Een functie wordt ook wel een *afbeelding* genoemd. We zeggen: een functie beeldt de elementen van een verzameling X af op de elementen van een verzameling Y. Een functie die alle elementen van X afbeeldt op elementen van Y heet *een functie van X naar Y*. Functies duiden we vaak aan met de letters f, g en h. "f is een functie van X naar Y" wordt vaak als volgt genoteerd:

$$f : X \to Y.$$

Een *functievoorschrift* beschrijft hoe een origineel wordt omgezet naar een beeld van dat origineel. Bijvoorbeeld: $a \mapsto a^2$.

Voorbeeld 3.19 Bekijk het functievoorschrift

$$f : \mathbb{Z} \to \mathbb{Z} : n \mapsto n^2.$$

Dit voorschrift geeft aan dat de functie f gehele getallen (weer terug-)afbeeldt op gehele getallen, en wel door individuele elementen af te beelden op hun kwadraten.

Opgave 3.11 Welke letters kunnen in het functievoorschrift hierboven worden veranderd zonder de betekenis van het voorschrift te veranderen?

Wanneer $f : X \to Y$ noemen we X het *domein* en Y het *co-domein* van f. De koppeling tussen domein en co-domein, $X \to Y$, noemen we de *signatuur* van een functie. Elk functievoorschrift móet worden voorafgegaan door een signatuur, anders is niet bekend waar de functie vandaan komt en waar die naar toe gaat. Merk op dat de gewone pijl, "$\to$," gebruikt wordt in de signatuur, terwijl de pijl met het linkerkantje, "$\mapsto$," gebruikt wordt in het daadwerkelijke functievoorschrift.

De dingen uit X waaraan door de functie f dingen uit Y worden toegekend heten de *argumenten* of *originelen* van de functie f. De dingen uit Y die door f aan dingen uit X worden toegekend heten de *beelden* of *waarden* van f.

We zeggen: het *toepassen* van de functie f op het argument a geeft als waarde het element b. Notatie: $f(a) = b$.

Laten we, voordat we naar wat exotischer functies gaan kijken, eerst nog even de officiële definitie van een functie geven. We kunnen namelijk een precieze verzamelingstheoretische definitie van het begrip "functie" geven:

Definitie 3.13 (Functie) *Laat X en Y verzamelingen zijn. Een* totale functie*, kortweg* functie *of* afbeelding

$$f : X \to Y$$

is een relatie f tussen X en Y, i.e., een deelverzameling $f \subseteq X \times Y$, zó dat voor elke $x \in X$ en $y, y' \in Y$ geldt:

$$\langle x, y\rangle \in f \text{ en } \langle x, y'\rangle \in f \qquad \Rightarrow \qquad y = y'$$

Met andere woorden: een functie is een twee-plaatsige relatie waarbij een linker-element nooit gerelateerd kan zijn aan twee verschillende rechter-elementen.

Het voorvoegsel "totaal" slaat op het gegeven dat $\mathrm{dom}(f) = X$, d.w.z. dat f gedefinieerd is voor elke $x \in X$. Naast totale functies bestaan er partiële functies. Deze worden elders in het dictaat besproken. Partiële functies komen minder voor. Als het woord "functie" valt, wordt daarom altijd "totale functie" bedoeld.

Opgave 3.12 Bekijk de functie

$$f : \mathbb{N} \to \mathbb{N} \times \mathbb{N} : n \mapsto (n+1, 2n).$$

1. Voor elke n lijkt f twee waarden te geven, namelijk $n+1$ en $2n$. Waarom is dit niet in strijd met Def. 3.13?
2. Van welke verzameling is f een deelverzameling als f beschouwd wordt als relatie?
3. Geef voorbeelden van programmeertalen waarbij het mogelijk is functies arrays te laten retourneren. Beschrijf hoe dit werkt.

Merkwaardige functies

Om vertrouwd te raken met randgevallen van functies en functievoorschriften gaan we een aantal merkwaardige functies bekijken.

Voorbeeld 3.20 (De Goldbachfunctie) Dit voorbeeld bouwt voort op het vermoeden van Goldbach (blz. 17). Bekijk het functievoorschrift

$$G : \mathbb{N} \to \{0,1\} : \quad n \mapsto \begin{cases} 1 & \text{als } n \text{ geschreven kan worden} \\ & \text{als de som van twee priemgetallen,} \\ 0 & \text{anders.} \end{cases}$$

Heel belangrijk is nu op te merken dat voor ieder argument, n, deze functie, zij het met enige moeite, is uit te rekenen: sommeer alle priemgetalparen kleiner dan n en kijk of je op enig moment success hebt. Dus hoewel tot op heden[1] het vermoeden van Goldbach ("$G(n) = 1$ voor alle even n") bewezen is, noch weerlegd, is de functiewaarde van G op elk argument gewoon uit te rekenen.

Het volgende voorbeeld toont aan dat het goed mogelijk is functies te definiëren waarvan de waarden bestaan, maar onbekend zijn. Het kostte ons enige moeite een dergelijke functie te vinden.[2]

Voorbeeld 3.21 (De Polignacfunctie) Een *priemhiaat ter grootte k* is een "gat" tussen twee priemgetallen ter grootte k. Bijvoorbeeld, het paar 31,37 is een priemhiaat ter grootte zes, want tussen 31 en 37 bevinden zich geen priemgetallen.

Bekijk de volgende functie.

$$H : \mathbb{N} \to \{0,1\} : \quad k \mapsto \begin{cases} 1 & \text{als er oneindig veel priemgetalhiaten,} \\ & \text{ter grootte } 2k \text{ bestaan,} \\ 0 & \text{anders.} \end{cases}$$

Deze functie is wel-gedefinieerd: voor vaste k bestaan er namelijk oneindig veel priemgetalhiaten ter grootte k, of niet. In 1849 publiceerde een Frans wiskundige Alphonse de Polignac (1817-1890) het volgende vermoeden:

> Voor elk natuurlijk getal k bestaan er oneindig veel priemgetalhiaten ter grootte $2k$.

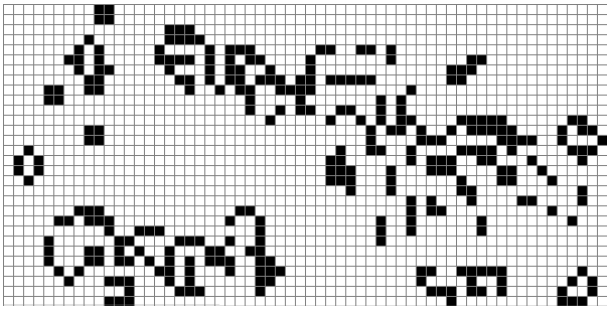
Tot op heden is voor elke k dit vermoeden bewezen noch weerlegd. De waarden van de functie H zijn tot nu toe dus onbekend. Merk op dat $H \equiv 1$ als en slechts als het vermoeden van De Polignac waar is. In het geval van $k = 1$ is het vermoeden van De Polignac gelijkwaardig aan het (dus ook onbewezen) *priemtweelingvermoeden*.

Voorbeeld 3.22 (De Game-of-lifefunctie) (Waarschuwing: in dit voorbeeld worden begrippen genoemd die later pas aan bod komen. Het betreft begrippen zoals "aftelbaar," "berekenbaar," en "halting problem".)

Laat

$$S = \{\text{ alle } \textit{eindige} \text{ deelverzamelingen van } \mathbb{Z} \times \mathbb{Z}\,\}.$$

Deze verzameling is (zoals we later zullen zien) aftelbaar en correspondeert eenduidig met start-configuraties in Conway's Game of Life. Hier zie je zo'n start-configuratie:



[1] 2018.

[2] Lees `http://mathoverflow.net/questions/107163/a-function-that-is-defined-everywhere-but-has-unknown-values` voor de ontstaansgeschiedenis van dit voorbeeld.

Laat $s \in S$ staan voor een start-configuratie. Bekijk de functie

$$\Omega : S \to \{0,1,2,3\} : s \mapsto \begin{cases} 0 & \text{als } s \text{ uitsterft,} \\ 1 & \text{als } s \text{ stabiliseert,} \\ 2 & \text{als } s \text{ zich gaat herhalen,} \\ 3 & \text{anders.} \end{cases}$$

Deze functie is wel-gedefinieerd, maar, zoals we later zullen zien, onberekenbaar (want equivalent aan de Halting functie). Dat Ω onberekenbaar is betekent dat er met zekerheid geen computerprogramma kan worden geschreven dat, voor een willekeurige start-configuratie $s \in S$, kan voorspellen of die configuratie uiteindelijk uitsterft, stabiliseert, zich gaat herhalen, of alsmaar groeit.

Opgave 3.13 1. Ga na dat S aftelbaar is.

2. Beredeneer dat Optie 3 equivalent is aan alsmaar groeien.

Voorbeeld 3.23 Om de randgevallen van het begrip functie verder te verkennen gaan we weer plaatjes tekenen. Hier is een (slechts bij grove benadering juist) plaatje van een werkloosheidsfunctie. Het plaatje laat het aantal ingeschreven werkloze academici per jaar zien, van 1978 tot en met 1987:[3]

1978	$\longrightarrow$	4200
1979	$\longrightarrow$	4800
1980	$\longrightarrow$	5200
1981	$\longrightarrow$	6400
1982	$\longrightarrow$	9000
1983	$\longrightarrow$	11500
1984	$\longrightarrow$	16000
1985	$\longrightarrow$	16400
1986	$\longrightarrow$	16000
1987	$\longrightarrow$	17000

Laten we deze werkloosheidsfunctie even g noemen. Uit het plaatje blijkt dat het domein van g de verzameling is van alle jaren vanaf 1979 tot en met 1987. Het co-domein van g is de verzameling van natuurlijke getallen. Uit het plaatje kunnen we bij voorbeeld aflezen welke waarde g toekent aan 1987: $g(1987) = 17000$. Merk op dat niet elk natuurlijk getal hoeft op te treden als waarde. Dat het co-domein van de functie de verzameling van natuurlijke getallen is wil alleen zeggen dat elke waarde die de functie kan aannemen een natuurlijk getal is. □

Voorbeeld 3.24 Nog een voorbeeld. Als we een verzameling mensen bekijken—noem die verzameling A—dan is "moeder van" een functie. Het is een manier om elk mens in A in verband te brengen met een element van een verzameling mensen B die alle moeders van mensen uit A bevat. Immers: gegeven een bepaalde persoon a, dan is er altijd een persoon te vinden die de moeder is van a. Hier is een voorbeeld-tabel:

Wilhelmina	$\longrightarrow$	Emma
Juliana	$\longrightarrow$	Wilhelmina
Beatrix	$\longrightarrow$	Juliana
Willem-Alexander	$\longrightarrow$	Beatrix

Noem deze functie h. Nu is h(Juliana) = Wilhelmina. De functie h kent aan argument Juliana de waarde Wilhelmina toe. Algemeen: de functie h kent aan elk element uit de verzameling

$$\{\text{Wilhelmina}, \text{Juliana}, \text{Beatrix}, \text{Willem-Alexander}\}$$

de moeder toe van dat element.

Abstract beschouwd is een functie met domein A en co-domein B niets anders dan een bijzondere vorm van een tweeplaatsige relatie binnen $A \times B$. Beschouw het voorbeeld van de koninklijke moeders: de functie uit de tabel kan als volgt worden genoteerd als een verzameling:

$$\{\langle \text{Wilhelmina}, \text{Emma}\rangle, \langle \text{Juliana}, \text{Wilhelmina}\rangle, \langle \text{Beatrix}, \text{Juliana}\rangle, \langle \text{Willem-Alexander}, \text{Beatrix}\rangle\}.$$

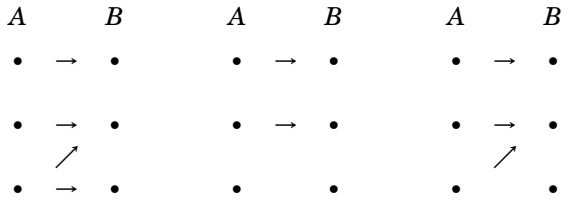
Deze verzameling is een tweeplaatsige relatie binnen het product:

$$\{\text{Wilhelmina}, \text{Juliana}, \text{Beatrix}, \text{Willem-Alexander}\} \times \{\text{Emma}, \text{Wilhelmina}, \text{Juliana}, \text{Beatrix}\}.$$

□

In het algemeen kunnen we een functie altijd beschouwen als een verzameling paren, waarbij de argumenten van de functie optreden als eerste lid van een paar, en de toegekende waarden als tweede lid. Dus: elke functie $f : A \to B$ is een relatie binnen $A \times B$.

Is nu ook elke relatie R binnen $A \times B$ een functie van A naar B? Nee, dat is niet zo, want het bijzondere aan een functie is, dat je voor elk argument een waarde moet kunnen aflezen. Met andere woorden: de waarde die aan een bepaald argument wordt toegekend moet *uniek zijn*. Verder moet bij een functie aan *elk* element uit A een element uit B worden toegekend. Bij een relatie hoeft dit niet. Enkele plaatjes ter verduidelijking:



De pijlen in de plaatjes links en midden geven een relatie tussen A en B aan, maar deze relatie is geen functie van A naar B. De reden in het plaatje links: sommige objecten uit A zijn gekoppeld aan meer dan één object in B. De reden in het middelste plaatje: niet alle objecten uit A hebben een waarde. De pijlen in het derde plaatje, tenslotte, geven een relatie binnen $A \times B$ aan die tevens een functie is van A naar B: elk element van A is uniek gekoppeld aan een element van B.

[3] De gegevens zijn afgelezen uit een grafiekje in *Carrière*, 12 december 1987.

Voorbeeld 3.25 De relatie tussen personen en hun geboorte-datum is een functie.

Voorbeeld 3.26 De relatie tussen personen en hun vaders is een functie.

Voorbeeld 3.27 De relatie tussen personen en hun broer is geen functie, want sommige mensen hebben geen broers, of ze hebben er meer dan één.

Voorbeeld 3.28 De relatie tussen personen en hun jongste zusje is geen functie, want sommige mensen hebben geen zusjes.

Voorbeeld 3.29 De relatie tussen landen en hun hoofdsteden is een functie (want elk land heeft precies één hoofdstad).

Voorbeeld 3.30 De relatie tussen landen en hun hoofdsteden annex residentiesteden is geen functie, want in sommige landen vallen hoofdstad en residentiestad niet samen.

Voorbeeld 3.31 De relatie tussen IBM-personal computers en hun serienummers is een functie, want elke PC heeft een uniek serienummer.

Voorbeeld 3.32 De relatie tussen natuurlijke getallen en hun kwadraten is een functie, want elk natuurlijk getal heeft een uniek kwadraat.

Voorbeeld 3.33 De relatie tussen natuurlijke getallen en hun derdemachtswortels is een functie.

Uit een relatie die geen functie is kan soms simpelweg een functie worden geconstrueerd door de domein-verzameling van de relatie anders te kiezen.

Voorbeeld 3.34 De relatie tussen mannen en hun huidige-echtgenotes in $M \times V$ (waarbij M de verzameling van alle mannen is, en V de verzameling van alle vrouwen) is geen functie, want sommige mannen zijn niet getrouwd. Wanneer we M inperken tot de verzameling M' van getrouwde mannen, dan is diezelfde relatie wel een functie: elke getrouwde man kan uniek worden afgebeeld op zijn huidige wettige echtgenote (we gaan hier even uit van een monogame samenleving).

Opgave 3.14 Ga na of de volgende relaties tevens functies zijn:

1. de relatie kind–ouder;
2. de relatie tussen een persoon en de verzameling van zijn/haar ouders.

Opgave 3.15 Herinner je dat de notie "functie" formeel als volgt is gedefinieerd:

> f is een **functie** van X naar Y dan en slechts dan als
>
> – $f \subseteq X \times Y$;
> – bij elke $x \in X$ is er precies één $y \in Y$ zodanig dat $\langle x, y\rangle \in f$.

(Herlees zo nodig Def. 3.13, blz. 34.) Ga na of er—volgens deze definitie—functies bestaan met als domein de lege verzameling. Zo nee, waarom niet? Zo ja, welke?

De *domein-* en *bereik*-definities voor relaties zijn nu ook van toepassing op functies. We hebben dus voor functie $f : X \to Y$ het volgende: $\mathrm{dom}(f) = X$ en $\mathrm{rng}(f) \subseteq Y$.

De verzameling van alle functies met domein X en co-domein Y wordt wel aangeduid als Y^X. Dus:

$$Y^X = \{f \subseteq X \times Y \mid f \text{ is een functie van } X \text{ naar } Y\}.$$

Opgave 3.16 Wat zijn de elementen van $\{c,d\}^{\{a,b\}}$?

Definitie 3.14 *Als f een functie is van X naar Y, en $A \subseteq X$, dan is $f[A]$,* **het beeld van** A **onder** f*, de verzameling*

$$\{y \in Y \mid \textit{er is een } a \in A \textit{ met } f(a) = y\}.$$

Opmerking: in veel teksten wordt het f-beeld van A genoteerd met ronde haken, als $f(A)$. Deze expressie is ambigu: er kan immers zowel $f[A]$ ("f op een verzameling") als $f[\{A\}]$ ("f op een individu") worden bedoeld. Meestal wordt f op een verzameling bedoeld. Het geval geval "f op een individu" komt bijna niet voor, omdat voor individuen bijna nooit hoofdletters worden gebruikt. Maak je geen zorgen: de betekenis van $f(A)$ volgt vrijwel altijd uit de context.

Hier is een opgave om te eventueel oefenen met het verschil tussen $f[A]$ en $f(A)$:

Opgave 3.17 Laat X de kleinste verzameling zijn die de lege verzameling bevat, plus al haar machtsverzamelingen. Dus:

a) $\emptyset \in X$

b) $A \subset X \Rightarrow A \in X$.

c) Niets anders is een element van X.

1. Laat zien dat $A = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \in X$.
2. Definieer $f : X \to X : x \mapsto |x|$. (De kardinaliteit, platweg: het aantal elementen, van de verzameling x.) Bereken $f[A]$. (We beschouwen A dus als een deelverzameling van X.)
3. Bereken $f(A)$. (We beschouwen A dus als een element van X.)

Definitie 3.15 *Als f een functie is van X naar Y, en $B \subseteq Y$, dan is $f^{-1}[B]$,* **inverse beeld** *of* **het volledig origineel van** B **onder** f*, de verzameling*

$$\{x \in X \mid \textit{er is een } b \in B \textit{ met } f(x) = b\}.$$

Uit deze definities volgt onmiddellijk dat als $f : X \to Y$, dan $f[X] = \text{rng}(f)$ en $f^{-1}[Y] = X$. (Dat laatste geldt alleen als f een totale functie is, maar dat wordt in het algemeen stilzwijgend aangenomen.)

Een functie f kan worden toegepast op een waarde van een functie g, mits de desbetreffende waarde van g maar in het domein van f ligt. In het algemeen: als $f : X \to Y$ en $f : Y \to X$, dan kunnen we, voor een $x \in X$, eerst x afbeelden op een element van Y met behulp van f, en vervolgens dit element van Y afbeelden op een element van Z met behulp van g. Het element van z waar we zo terecht komen kan worden aangeduid als $g(f(x))$. Deze procedure van eerst f en vervolgens g toepassen levert ons in feite een *nieuwe* functie op, een functie met domein X en co-domein Z. Deze functie wordt de *compositiefunctie* $g \circ f$ (spreek uit "g na f") genoemd. We kunnen $g(f(x))$ nu ook noteren als $g \circ f(x)$. Wederom: plaatjes kunnen dit verduidelijken:

$$\begin{array}{ccccc} X & \overset{f}{} & Y & \overset{g}{} & Z \\ x & \to & y & \to & z \end{array} \qquad \begin{array}{ccc} X & \overset{g\circ f}{} & Z \\ x & \to & z \end{array}$$

Een precieze definitie:

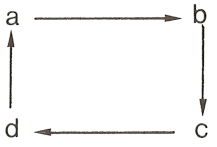
Definitie 3.16 (Functiecompositie) *Zij $f : X \to Y$ en $g : Y \to X$ functies. De functie $g \circ f$ is gedefinieerd als de relatie*

$$g \circ f =_{Def} \{\langle x, z\rangle \in X \times Z \mid g(f(x)) = z\}.$$

Wanneer f een functie is met een en dezelfde verzameling X als domein en als co-domein, dan kunnen we de functie op waarden van zichzelf toepassen. Dit heet *functie-iteratie*. Voorbeeld: de moeder van de moeder van de moeder van Willem Alexander. Of: het kwadraat van het kwadraat van 2.

Definitie 3.17 *Een functie $X \to X$ wordt een* **éénplaatsige operator** *op X genoemd. Een functie $X^n \to X$ wordt een n-***plaatsige operator** *op X genoemd.*

Opgave 3.18 Beschouw het volgende plaatje.



De relatie die getekend is geeft voor ieder element van A precies één ding waartoe dat element in relatie staat. Ditzelfde in andere woorden gezegd: de relatie is *functioneel*. Noem deze functionele relatie (dat wil zeggen: functie) voor het gemak f.

1. Bepaal $f(a)$.
2. Bepaal $f(f(a))$.
3. Bepaal $f(f(f(a)))$.
4. Bepaal $f(f(f(f(a))))$.
5. Bepaal $f(b)$.
6. Bepaal $f(f(f(f(b))))$.

Opmerking: $f(f(a))$ kan ook worden geschreven als: $f \circ f(a)$, $f(f(f(a)))$ als $f \circ f \circ f(a)$, enzovoorts.

Opgave 3.19 Breid het plaatje uit de vorige opdracht uit met pijlen die de functionele relatie $f \circ f$ aanduiden (gebruik pijlen van de vorm $\Longrightarrow$).

Merk op dat het domein van een functie f zelf een Cartesisch product kan zijn. In dat geval bestaan de argumenten van f uit rijtjes. Laat $f : A^2 \to B$ zo'n functie zijn. Dan kent f aan elk paar $\langle x, y\rangle \in A^2$ een element van B toe. In plaats van $f(\langle x, y\rangle)$ schrijft men meestal: $f(x, y)$. De functie f kent waarden toe aan tweetallen argumenten.

Een functie $f : A^2 \to A$ wordt een tweeplaatsige operatie op A genoemd; een functie $f : A^3 \to A$ heet een drieplaatsige operatie op A, enzovoorts.

Voorbeeld 3.35 De functie + die aan elk paar van natuurlijke getallen $\langle m, n\rangle$ het getal $m + n$ toekent is een tweeplaatsige operatie op $\mathbb{N}$. Deze operatie wordt in de wandeling 'optellen' genoemd. Merk op dat de optel-operatie *commutatief* is: de volgorde waarin de argumenten worden genomen maakt niet uit.

We besluiten met nog wat notatie. Als $f : \mathbb{N} \to \mathbb{N}$ (dat wil zeggen f is een functie met als domein en als co-domein de natuurlijke getallen), en f is de functie die ieder getal afbeeldt op zijn kwadraat (anders gezegd: f is de functie 'kwadrateren'), dan drukken we dit wel uit met behulp van het voorschrift $f(x) = x^2$. De functie $g : \mathbb{N} \to \mathbb{N}$ die elk getal eerst kwadrateert en er dan 1 bij optelt heeft als voorschrift $g(x) = x^2 + 1$. Als h staat voor de functie die twee natuurlijke getallen bij elkaar optelt en het resultaat vervolgens kwadrateert, dan wordt het functie-voorschrift:

$$h(x, y) = (x + y)^2.$$

In feite legt zo'n functie-voorschrift uit wat een functie doet in termen van operaties die al eerder gegeven zijn en waarvan de werking bekend wordt verondersteld, zoals 'optellen' en 'machtsverheffen'.

Definitie 3.18 *Zij $f : X \to Y$ en $g : X \to Y$ twee functies. We noemen f en g* identiek*, en schrijven $f \equiv g$, als voor alle $x \in X$: $f(x) = g(x)$.*

Twee functies zijn dus identiek als en slechts ze op alle elementen in het domein hetzelfde "doen".

Later zal overigens het symbool "$\equiv$" ook nog worden gebruikt om de logische equivalentie van twee uitspraken aan te duiden. Daarmee wordt dit symbool overladen (Eng.: *overloaded*) en moeten we uit de context opmaken waar "$\equiv$" voor staat. Bijna nooit blijkt dat een probleem te zijn.

Opgave 3.20 Laat $f : A \to B$; $g : B \to C$; $A', A'' \subseteq A$; $B', B'' \subseteq B$; $C' \subseteq C$ willekeurig. Bewijs of weerleg:

1. $f[A' \cap A''] = f[A'] \cap f[A'']$.
2. $f[A' \cup A''] = f[A'] \cup f[A'']$.
3. $f[A' \setminus A''] = f[A'] \setminus f[A'']$.
4. $f^{-1}[B' \cap B''] = f^{-1}[B'] \cap f^{-1}[B'']$.
5. $f^{-1}[B' \cup B''] = f^{-1}[B'] \cup f^{-1}[B'']$.
6. $f^{-1}[B' \setminus B''] = f^{-1}[B'] \setminus f^{-1}[B'']$.
7. $f^{-1}[f[A']] \subseteq A'$.
8. $f^{-1}[f[A']] \supseteq A'$.
9. $f[f^{-1}[B']] \subseteq B'$.
10. $f[f^{-1}[B']] \supseteq B'$.
11. $g \circ f[A'] = g(f[A'])$.
12. $(g \circ f)^{-1}[C'] = f^{-1}[g^{-1}[C']]$.

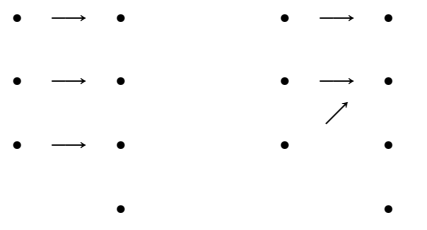
3.3 Injectie, surjectie, bi-jectie

Functies kunnen injectief, surjectief, of allebei (bi-jectief) zijn. Deze eigenschappen komen voortdurend terug.

Definitie 3.19 *Een functie f heet* **injectief** *of* **één-één-duidig** *of* **1-op-1** *als f aan verschillende elementen uit zijn domein verschillende waarden toekent.*

Iets formeler: als $a \neq b$ dan $f(a) \neq f(b)$. Of, wat hetzelfde is: als $f(a) = f(b)$ dan $a = b$. (De laatste definite is iets puurder, omdat deze niet met ontkenningen werkt.)

In een paar plaatjes:



De functie uit het plaatje links is injectief. De functie uit het plaatje rechts is dat niet. Een injectieve functie wordt ook wel een *injectie* genoemd. De notatie voor 'f is een injectie van A naar B' is: $f : A \xrightarrow{i} B$.

Opgave 3.21 Een zekere verzameling A heeft 15 elementen. Verder is gegeven dat er een injectieve functie $f : A \to B$ bestaat. Wat kun je hieruit afleiden met betrekking tot het aantal elementen in B?

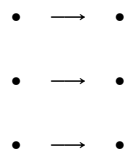
Definitie 3.20 *Een functie f heet* **surjectief** *wanneer elk element uit het co-domein van f optreedt als functie-waarde.*

Iets formeler: als f een functie is van A naar B, dan is er voor elke $b \in B$ een $a \in A$ met $b = f(a)$. Dus: een functie $f : A \to B$ is surjectief desda $B = \text{rng}(f)$. We verduidelijken één en ander weer met plaatjes:

De functie uit het plaatje links is surjectief; de functie uit het plaatje rechts is dat niet. Een surjectieve functie wordt wel een *surjectie* genoemd. Notatie voor surjecties: $f : A \xrightarrow{s} B$. Voor een surjectie zegt men ook wel: f is een functie van A **op** B (Eng.: *onto*).

Definitie 3.21 *Een functie f heet* **bi-jectief** *of een* **1-op-1 correspondentie** *wanneer f zowel injectief als surjectief is.*

Een kortere benaming voor bi-jectieve functie is *bi-jectie*. Notatie voor bi-jecties: $f : A \xrightarrow{i,s} B$. Een plaatje van een bi-jectie:



Als f een bi-jectie is van A naar B dan brengt f een één–één-duidige correspondentie tot stand tussen de leden van A en de leden van B: bij iedere $a \in A$ hoort precies één $b \in B$ met $b = f(a)$, en bij iedere $b \in B$ hoort precies één $a \in A$ met $b = f(a)$ (ga dit zelf na aan de hand van de eigenschappen van een bi-jectie).

Opgave 3.22 $\mathbb{N}$ is de verzameling van natuurlijke getallen (de verzameling $\{0, 1, 2, 3, ...\}$, en $f : \mathbb{N} \to \mathbb{N}$ is de functie 'met 2 vermenigvuldigen' (dat wil zeggen $f(0) = 0$, $f(1) = 2$, $f(2) = 4$, enzovoorts). Is f surjectief? Injectief? Bi-jectief?

Als f een bi-jectie is, dan kunnen we de richting van de functie omdraaien door eenvoudigweg elk element $\langle a, b \rangle$ van f te veranderen in $\langle b, a \rangle$. Het resultaat zal weer een functie zijn, en wel: een bi-jectie. Immers, omdat f bi-jectief is, is er voor elke waarde van f een uniek argument dat die waarde oplevert. De functie die door zo uit een bi-jectie f ontstaat door 'omdraaien' heet de *inverse functie* van f, en wordt genoteerd als: f^{-1}. Dus: als $f : A \to B$ een bi-jectie is, dan is f^{-1} de functie van B naar A die aan elke $b \in B$ het uniek bepaalde element $a \in A$ koppelt waarvoor geldt dat $f(a) = b$. Preciezer:

Definitie 3.22 (Inverse functie) *Zij $f : A \to B$ een bi-jectie. De* inverse functie $f^{-1} : B \to A$ *is gedefinieerd als de relatie*

$$f^{-1} =_{Def} \{\langle b, a \rangle \in B \times A \mid f(a) = b\}.$$

Voorbeeld 3.36 De functie f uit het linkerplaatje is een bi-jectie, en dus is ook f^{-1} uit het rechterplaatje een bi-jectie:

A	f	B
a	$\longrightarrow$	w
b	$\longrightarrow$	x
c	$\longrightarrow$	y
d	$\longrightarrow$	z

B	f^{-1}	A
w	$\longrightarrow$	a
x	$\longrightarrow$	b
y	$\longrightarrow$	c
z	$\longrightarrow$	d

Voorbeeld 3.37 De functie

$$f : \mathbb{N} \to \{x \mid x \in \mathbb{N} \text{ en } x \text{ is even}\}$$

die gedefinieerd wordt door $f(n) = 2n$ is een bi-jectie. De inverse functie

$$f^{-1} : \{x \mid x \in \mathbb{N} \text{ en } x \text{ is even}\} \to \mathbb{N}$$

is gedefinieerd door $f^{-1}(m) = m/2$.

Hoewel niet elke functie een inverse bezit, kan er *altijd* gesproken worden over het inverse beeld.

Voorbeeld 3.38 De functie

$$f : \mathbb{Z} \to \mathbb{Z} : x \mapsto x^2$$

bezit geen inverse, want f is niet bi-jectief. Het heeft dus geen zin om over $f^{-1}(4)$ te praten, want f^{-1} bestaat niet.

Echter, het inverse beeld van 4 onder f bestaat wel: $f^{-1}[4] = \{-2, 2\}$. De rechthoekige haken geven aan dat het om het inverse beeld (van 4) gaat, en niet om het beeld (van 4) van de inverse.

Dus: een inverse (en dus het beeld van de inverse) bestaat niet altijd, maar een inverse beeld bestaat altijd.

Waarschuwing: In veel teksten wordt het f-inverse beeld van B' genoteerd met ronde haken, als $f^{-1}(B')$. Deze expressie is ambigu: er kan immers ook het beeld van B' onder de inverse van f bedoeld worden (aangenomen dat de laatste bestaat en welgedefinieerd is). Laten we dit controleren middels een opgave:

Opgave 3.23 Laat $f : X \to Y$ een functie zijn waarvan de inverse bestaat. Laat zien dat voor elke $B \subseteq Y$:

$$f^{-1}(B) \text{ bestaat en } f^{-1}(B) = f^{-1}[B]$$

Opgave 3.24 Bewijs:

1. Als er een injectie is van A naar B, dan is er een surjectie van B naar A.
2. Als er een surjectie is van A naar B, dan is er een injectie van B naar A.

In het bewijs van beide onderdelen maak je (hoogstwaarschijnlijk onbewust) gebruik van het keuzeaxioma (blz. 51). Geen punt. Lees de sectie over het keuzeaxioma later nog eens een keer door.

3.4 Variaties op functies

Een bestaande functie kan uitgebreid, beperkt, of geprojecteerd worden. Op zichzelf genomen is dit niet bijster spannend, maar wel heel erg nodig voor bijzonder veel andere onderwerpen. Je kunt overwegen dit gedeelte pas te bestuderen als je het nodig hebt.

Uitbreidingen en restricties

Uitbreiding en restrictie zijn tegenovergestelde begrippen.

Definitie 3.23 *Zij $A \subseteq X$, $f : A \to Y$ en $g : X \to Y$. Als voor elke $a \in A$ geldt dat $g(a) = f(a)$, dan zeggen we dat g een* uitbreiding *is van f naar X; we zeggen tegelijkertijd dat f een* beperking *of* restrictie *van g is op A. In beide gevallen noteren we dit als*

$$f = g|_A.$$

Als g en h identiek zijn op A, i.e., $g|_A \equiv h|_A$, dan wordt soms, om een extra "A" te vermijden, $g \equiv_A h$ geschreven.

Merk op dat, per definitie, f en $g|_A$ identiek zijn: $f \equiv g|_A$. (Zie zo nodig Definitie 3.18 op blz. 37.) Soms wordt ook $g =_A h$ geschreven i.p.v. $g \equiv_A h$.

Opgave 3.25 1. Laat $A \subseteq B \subseteq X$ en $f : A \to Y$, $g : B \to Y$, $F : X \to Y$, zó dat $g|_A \equiv f$ en $F|_B \equiv g$. Je raad het al: we vragen je te laten zien dat $F|_A \equiv f$.

2. Zij X en Y verzamelingen met respectievelijk $|X| = m$, $|Y| = n$, waarbij m en n geheel en niet negatief.
   (a) Hoeveel elementen bevat Y^X?
   (b) Laat $A \subseteq X$ met $|A| = r$ en $0 \leq r < m$ geheel, en laat $f : A \to Y$. Op hoeveel verschillende manieren kan f naar X worden uitgebreid?

Projecties

Een projectie licht één of meerdere factoren uit een Cartesisch product. Projecties spelen bijvoorbeeld een rol in de berekenbaarheidstheorie.

Definitie 3.24 *Zij $A_1 \times A_2$ een Cartesisch product. De i^e* projectie *is de functie*

$$\pi_i : X_1 \times X_2 \to X_i : (x_1, x_2) \mapsto x_i.$$

Voorbeeld 3.39 De projectie op de tweede coördinaat van $\mathbb{N} \times \mathbb{Q}$, genoteerd als π_2, beeldt bijvoorbeeld $(4, 2.5)$ af op 2.5. En bijvoorbeeld $(9, -3.2)$ op -3.2.

Opmerkingen:

– Projecties worden bijna altijd gebruikt in combinatie met andere functies, bijvoorbeeld eerst een projectie $\mathbb{N}^2 \to \mathbb{N}$ gevolgd door een functie $\mathbb{N} \to \mathbb{N}$.

– Veel programmeertalen kennen het begrip projectie ook. Denk aan Haskell en andere functionele talen.

– De notie van een projectie kan worden gegeneraliseerd naar aftelbaar oneindig en over-aftelbare Cartesische producten. Als I een index-verzameling is, dan is

$$\pi_i : \prod_{a \in I} A_i : \bar{x} \mapsto x_i$$

net zo goed een projectie.

Opgave 3.26 Laat zien dat projecties altijd surjectief zijn.

Opgave 3.27 Zij $\{A_i\}_{i \in I}$ en $\{B_i\}_{i \in I}$ twee families van verzamelingen met dezelfde indexerings-verzameling I.

1. Stel, om het een beetje concreet te maken, dat $I = \{1,2,3,4,5\}$, $A_i = \{x \in \mathbb{N} \mid x \leq i\}$ en $B_i = \{x \in \mathbb{N} \mid x \geq i\}$. Geef van de volgende tupels aan of ze tot het product

   $$A = \prod_{i \in I} A_i \, (= A_1 \times \cdots \times A_5)$$

   behoren. Geef vervolgens aan of tupels tot het product $B = \prod_{i \in I} B_i$ behoren.

   $(1,2,3,4,5)$, $(1,1,1,1,1)$, $(5,5,5,5,5)$, $(0,0,0,0,0)$, $(9,9,9,9,9)$, $(1,2,2,2,2)$, $(3,3,3,4,5)$, $(5,4,3,2,1)$.

2. Zij

   $$f_i : A_i \to B_i : x \mapsto 2i - x.$$

   Vul een paar waarden in. Wat doet elke f_i? Beargumenteer dat elke functie f_i welgedefinieerd is (i.e., nooit "crasht").

3. Toon aan dat er precies één functie $f : A \to B$ bestaat, zó dat $p_i \circ f = f_i \circ p_i$, voor elke $i \in I$, waarbij p_i de passende projectie is. Deze functie wordt genoteerd als $\prod_{i \in I} f_i$.

   Geef de signaturen (domein, bereik, functievoorschrift) van $p_i \circ f$ en $f \circ p_i$.

4. De aannamen in onderdeel 1 komen te vervallen. Stel dat er voor elke $i \in I$ een functie $f_i : A_i \to B_i$ is. Toon aan dat $\prod_{i \in I} f_i$ bestaat.

5. Stel dat $\{C_i\}_{i \in I}$ gegeven is, en een tweede familie van functies $g_i : B_i \to C_i$. Toon aan dat $\prod_{i \in I} g_i \circ \prod_{i \in I} f_i = \prod_{i \in I} g_i \circ f_i$.

6. Stel dat elke f_i een inverse f_i^{-1} bezit. Geef de inverse van $\prod_{i \in I} f_i$.

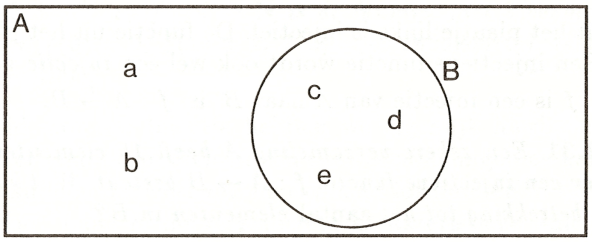
3.5 Bijzondere functies

Functies die als co-domein de verzameling $\{0,1\}$ hebben noemen we *karakteristieke functies*. Een karakteristieke functie $f : A \to \{0,1\}$ doet in feite niets anders dan een *deelverzameling* van A karakteriseren, namelijk de deelverzameling van die elementen van A die op 1 worden afgebeeld. Weer kan een tabel helpen:

A	f	$\{0,1\}$
a	$\longrightarrow$	1
b	$\longrightarrow$	0
c	$\longrightarrow$	1
d	$\longrightarrow$	1
e	$\longrightarrow$	0

De functie f uit het plaatje karakteriseert de verzameling $\{a,c,d\} \subset A$.

Opgave 3.28 Het volgende plaatje laat een verzameling A met een echte deelverzameling B zien.



Som de elementen op van de functie $f : A \to \{0,1\}$ die B karakteriseert.

Er bestaat een één–één correspondentie tussen karakteristieke functies met domein A en deelverzamelingen van A. Het verband is als volgt:

– Met de karakteristieke functie $f : A \to \{0,1\}$ correspondeert de volgende deelverzameling A' van A:

  $A' = \{a \in A \mid f(a) = 1\}$.

– Als $A' \subseteq A$ dan is de karakteristieke functie $f : A \to \{0,1\}$ die A' karakteriseert de volgende functie:

  $f = \{\langle a, 1\rangle \mid a \in A'\} \cup \{\langle a, 0\rangle \mid a \in A - A'\}$.

Opgave 3.29 Ga na dat $\{\langle a,1\rangle \mid a \in A'\} \cup \{\langle a,0\rangle \mid a \in A - A'\}$ inderdaad

1. een functie is,
2. een karakteristieke functie is,
3. de karakteristieke functie is die inderdaad A' karakteriseert.

Definitie 3.25 (Identiteitsfunctie) *Laat A een verzameling zijn. De* identiteitsfunctie *op A, genoteerd als id of, als we willen laten zien dat id op A werkt, id_A, is de functie van A naar A die ieder element op zichzelf afbeeldt:*

$$id_A : A \to A : x \mapsto x.$$

Opgave 3.30 Bewijs dat voor elke verzameling de identiteitsfunctie injectief, surjectief en bi-jectief is. Laat zien dat de identiteitsfunctie uniek is.

Definitie 3.26 (Inbedding) *Laat $A \subseteq B$. De* inbedding *van A in B, genoteerd als ι of, als we willen laten zien dat ι op A werkt, ι_A, is de functie van A naar B die ieder element op zichzelf afbeeldt:*

$$\iota_A : A \to B : x \mapsto x.$$

Opgave 3.31 Laat zien dat elke inbedding injectief is. Laat ook zien dat niet elke inbedding surjectief is.

Opgave 3.32 Zij $f : A \to B$ gegeven. Bewijs:

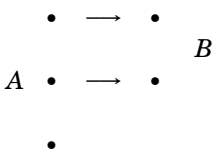
1. f is injectief $\Leftrightarrow$ f bezit een *links-inverse*, i.e., een functie $g : B \to A$ zodanig dat $g \circ f = \mathrm{id}_A$.
2. f is surjectief $\Leftrightarrow$ f bezit een *rechts-inverse*, i.e., een functie $g : B \to A$ zodanig dat $f \circ g = \mathrm{id}_B$.
3. f is bi-jectief $\Leftrightarrow$ f bezit een *inverse*, i.e., een functie $g : B \to A$ zodanig dat $f(x) = y \Leftrightarrow g(y) = x$, voor alle $x \in A$, $y \in B$.
4. Bewijs dat een inverse functie uniek is, en dat we dus mogen spreken over *de* inverse functie.
5. Bewijs dat links- en rechtsinverses niet uniek zijn.

We zullen in het volgende hoofdstuk zien dat bi-jecties een grote rol spelen bij het redeneren over oneindige verzamelingen. De meeste verzamelingen die we je hierboven hebben voorgeschoteld waren eindig, maar dat was alleen om didactische redenen. Verzamelingenleer gaat pas echt geestverruimend werken wanneer we de innerlijke blik richten op het oneindige. In Hoofdstuk 4 nemen we je mee om een kijkje te nemen in 'Cantor's paradijs' van oneindige verzamelingen.

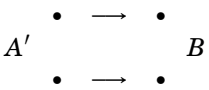
Partiële functies

Een relatie die strikt genomen geen functie is omdat zij niet voor elk argument een waarde oplevert, maar die bij inperking van het domein *wel* een functie is voor dat nieuwe domein wordt wel een *partiële functie* genoemd.

Voorbeeld 3.40 Hier is een abstract voorbeeld van een partiële functie:



Inperking van het domein geeft:



Voorbeeld 3.41 De functie $f : \mathbb{R} \to \mathbb{R} : x \to 1/x$ is een concreet voorbeeld van een partiële functie. Immers, voor het getal nul is de functie niet gedefinieerd.

Partiële functies worden vooral gebruikt als we met functies werken waarvan het exacte domein vooraf niet helemaal duidelijk is. In de berekenbaarheidstheorie, bijvoorbeeld, koppelen we functies aan programma's, en voor sommige programma's is niet voor elke invoer van tevoren duidelijk of voor die invoer het resultaat gedefinieerd is.

Laten we, ook voor later gebruik, een nette definitie geven:

Definitie 3.27 (Partiële functie) *Een* partiële functie *$f : A \to B$ is een relatie $f \subseteq A \times B$ tussen twee verzamelingen, die vanuit A gezien, bijna een functie is. Dat wil zeggen, er is een niet-lege deelverzameling $A' \subseteq A$, zó dat $f|_{A'} \subseteq A \times B$ wél een functie is.*

Als f gedefinieerd is op x, schrijven we $f \downarrow x$ (geheugensteun: híer [vinger prikt op de tafel] is de functie gedefinieerd). Als f niet gedefinieerd is op x, schrijven we $f \uparrow x$ (geheugensteun: de waarde van deze functie is hier vervlogen). We geven toe dat de tweede geheugensteun wat potsierlijk aandoet, maar samen werken ze bij ons in ieder geval prima.

Een *totale functie* is een functie zoals we die kennen, dat wil zeggen, een functie die op alle waarden in het domein gedefinieerd is. Elke totale functie is ook een (oneigenlijke) partiële functie.

We spraken net wat achteloos over het *domein* van een partiële functie. Helaas is het domein van een partiële functie in de wiskunde niet duidelijk gedefinieerd. Met andere woorden, er is helaas niet goed afgesproken wat men onder het domein van een partiële functie verstaat. Er zijn tenminste twee definities. Laat $f : A \to B$ een partiële functie zijn en laat

$$A' = \{a \in A \mid f \downarrow a\}.$$

het gedeelte zijn waar f is gedefinieerd.

Versie 1. Het *domein* van een partiële functie $f : A \to B$ is A. De verzameling A' wordt dan het *domein van definitie* genoemd, en genoteerd als $\mathrm{domdef}(f)$.

Versie 2. Het *domein* van $f : A \to B$ is A'. Een probleem met deze definitie is dat er nu geen naam meer is voor de relatie tussen f en A, immers de naam "domein" is al bezet. Toch wordt deze definitie nog wel gebruikt, met name in de berekenbaarheidstheorie. Als $A' \neq A$ wordt A heel soms de *bron* (Eng.: *source*) genoemd.

Elke niet-totale partiële functie is overigens altijd op een voor de hand liggende manier uit te breiden naar een totale functie door een willekeurig element b in het co-domein te nemen en dan $f(a) = b$ te definiëren voor alle waarden van a waar f eerst ongedefinieerd was. Soms wordt het op een andere manier opgelost, dan wordt er aan het co-domein een extra element, meestal $\perp$ ("undefined") toegevoegd en $f(a) = \perp$ gedefinieerd voor alle waarden van a waar f eerst ongedefinieerd was.

Opgave 3.33 Geef voor elk van de volgende functies het domein (versie 1), het domein (versie 2), het domein van definitie, de bron, en het bereik.

a) $f : \mathbb{R} \to \mathbb{R} : x \mapsto \sqrt{x}$

b) $f : \mathbb{R} \to \mathbb{R} : x \mapsto \log(x)$

c) $f : \mathbb{R} \to \mathbb{R} : x \mapsto 1/(2 - x^2)$

d) $f : \mathbb{R} \to \mathbb{R} : x \mapsto 1/\sin(x)$

e) $f : \mathbb{R} \to \mathbb{R} : x \mapsto 1/\sin(1/x)$

f) $f : \mathbb{N} \to \mathbb{N} : x \mapsto 95 - x^2$

Opgave 3.34 Laat de volgende programma's gegeven zijn, waarbij ':=' de toekenningsoperator voorstelt en '%' de rest-bij-deling operator:

$P_1(x)$: **while** $(x > 10)$ { $x := x + 1$ } **return** x;

$P_2(x)$: **while** $(x \neq 0)$ { $x := (x + 2)\%3$ } **return** x;

$P_3(x)$: **while** $(x \neq 0)$ { $x := (x + 2)\%4$ } **return** x;

$P_4(x)$: **while** $(x \neq 0)$ { $x := (x^2)\%2$ } **return** x;

Geef voor elk van de volgende functies het domein (versie 1), het domein (versie 2), het domein van definitie, de bron en het bereik.

a) $f : \mathbb{N} \to \mathbb{N} : x \mapsto P_1(x)$

b) $f : \mathbb{N} \to \mathbb{N} : x \mapsto P_2(x)$

c) $f : \mathbb{N} \to \mathbb{N} : x \mapsto P_3(x)$

d) $f : \mathbb{N} \to \mathbb{N} : x \mapsto P_4(x)$

Opgave 3.35 Totale functies bezitten een inverse als en alleen als ze bi-jectief zijn.

1. Bewijs dat partiële functies een inverse bezitten als en alleen als ze injectief zijn.
2. Laat zien dat als de oorspronkelijk functie niet totaal is, de inverse functie dat ook niet is.

Voor sommige functies is het niet makkelijk danwel onmogelijk uit te maken of zij totaal zijn. Voor bijvoorbeeld de Collatz-functie

$$f : \mathbb{N} \to \{\text{halt}\} : n \mapsto \begin{cases} \text{halt} & \text{als } n \leq 1, \\ f(n/2) & \text{als } n \text{ even is,} \\ f(3n+1) & \text{anders.} \end{cases}$$

is tot op heden[4] onbekend of deze totaal is.

Merk overigens op dat de functie-waarden van deze functie niet interessant zijn. Immers, er is maar één mogelijke functiewaarde, te weten: "halt". Het gaat er bij de Collatz-functie juist om voor welke waarden uit $\mathbb{N}$ de functie f gedefinieerd is. Voor meer uitleg over de Collatz-functie verwijzen we naar blz. 214.

[4] 2018.

Opgave 3.36 Iemand beweert dat de Collatz-functie partieel is. Geef je commentaar.

Opgave 3.37 Een *lineaire congruentie generator* (LCG)

$$x_i = \begin{cases} s & \text{als } i = 0, \\ ax_{i-1} + c \pmod m & \text{anders.} \end{cases}$$

is één van de oudste manieren pseudo-toevalsgetallen $x_1, x_2, x_3, x_4, \ldots$ te genereren. De constanten s (seed), a, c en m (modulus) zijn parameters die verstandig moeten worden gekozen, omdat er anders toch weer te veel regelmatigheid in de gegenereerde getallen zitten.

De periode van een LCG is hoogstens m en voor sommige keuzes veel kleiner. Men kan bewijzen dat een LCG een volledige cyclus doorloopt voor alle s als en slechts als $c \neq 0$ en

1. c en m zijn relatief priem (i.e. grootste gemene deler is 1)
2. $a - 1$ is deelbaar door alle priemfactoren van m
3. $a - 1$ is een viervoud als m een viervoud is

Een veel geciteerde keuze is die van Lewis, Goodman en Miller, die $a = 7^5 = 16807$, $c = 0$, $m = 2^{31} - 1 = 2147483647$ voorstelden.

Voor elke a, c, m, en k is er het volgende programma $P_{a,c,m,k}(x)$

$x := s$;
{
  Itereer de LCG met parameters a, c, m.
  Stop als $x = k$.
}

Is de functie die hoort bij $P_{a,c,m,k}$ met a, c, m de parameters van Lewis, Goodman en Miller en $k = 16 \pmod m$ totaal?

Hoofdstuk 4

Oneindigheid

4.1 Eindig versus oneindig

Intuïtief gesproken is een eindige verzameling een verzameling die de eigenschap heeft dat we de elementen van die verzameling kunnen tellen, en wel zo dat dat telproces op een gegeven ogenblik is afgerond. Let wel: het gaat er hier om dat het telproces *in principe* kan worden afgesloten. Eindige verzamelingen kunnen welhaast onvoorstelbaar groot zijn, zonder dat dit iets toe of af doet aan hun eindigheid.

Volgens het Boeddhisme kost het doorlopen van de cirkel van wedergeboorten, nodig om de Verlichting te bereiken, aeonen van tijd. Vraag aan de Boeddha: hoe lang duurt een aeon? Antwoord: "Stel je een machtige rotsformatie voor van vier mijlen hoog, breed en diep, een volkomen massief blok zonder een barst of scheurtje. Stel dat er aan het eind van iedere eeuw een man zou komen die met een doek van Benares eenmaal langs de rots zou strijken. Die bergrots zou dan eerder zijn weggesleten dan er een aeon voorbij is." Zie [Humphreys 1951].

Bevat een aeon nu oneindig veel seconden? Nee. Wel onvoorstelbaar veel, maar dat is niet hetzelfde. Het aantal seconden in een aeon is eindig, want hoewel een aeon onvoorstelbaar lang duurt, hij is op een gegeven moment verstreken. Dus: er bestaat een natuurlijk getal (zij het astronomisch groot) dat dat aantal seconden van een aeon aangeeft.

Goed, je hebt nog wel even tijd om aan logica en formele taalkunde te besteden voordat je de Verlichting bereikt. Een formele definitie van *eindigheid* en *oneindigheid* maakt gebruik van het begrip *bi-jectie* (zie § 3.5). Een bi-jectie of één–één-correspondentie tussen een verzameling A en een verzameling B is een functie f zo dat bij iedere $a \in A$ precies een $b \in B$ zo dat $f(a) = b$, en bij iedere $b \in B$ precies een $a \in A$ zo dat $b = f(a)$. Door middel van een bi-jectie kunnen we nu een *willekeurige* eindige verzameling in verband brengen met een *speciaal* soort eindige verzameling. Voor elke verzameling met n elementen is er immers een 1–1-correspondentie met de verzameling $\{0, 1, \ldots, n-1\}$. Een dergelijke verzameling wordt wel een *echt beginstuk* van $\mathbb{N}$ genoemd. Een meer exacte en algemene notatie hiervoor is $\{x \in \mathbb{N} \mid x < n\}$.

Definitie 4.1 *Verzameling A heet* **eindig** *wanneer er een $n \in \mathbb{N}$ te vinden is zo dat er een bi-jectie bestaat tussen $\{x \in \mathbb{N} \mid x < n\}$ en A.*

Voorbeeld 4.1 $\emptyset$ is eindig. Waarom? Omdat er een bi-jectie bestaat tussen de verzameling $\{x \in \mathbb{N} \mid x < 0\}$ en de lege verzameling.

Voorbeeld 4.2 De verzameling $\{a, b, c\}$ is eindig. De volgende functie is immers een bi-jectie tussen $\{x \in \mathbb{N} \mid x < 4\}$ en $\{a, b, c\}$:

$$\begin{array}{ccc} 1 & \longrightarrow & \text{a} \\ 2 & \longrightarrow & \text{b} \\ 3 & \longrightarrow & \text{c} \end{array}$$

Voorbeeld 4.3 De verzameling $\{1, 3, 5, 7, 9\}$ is eindig. De volgende functie is een bi-jectie tussen de verzameling $\{x \in \mathbb{N} \mid x < 6\}$ en deze verzameling:

$$\begin{array}{ccc} 1 & \longrightarrow & 1 \\ 2 & \longrightarrow & 3 \\ 3 & \longrightarrow & 5 \\ 4 & \longrightarrow & 7 \\ 5 & \longrightarrow & 9 \end{array}$$

Het zal duidelijk zijn dat het aanbrengen van een bi-jectie tussen een verzameling A en een beginstuk van de natuurlijke getallen niets anders is dan wat wij in het alledaagse taalgebruik 'tellen' noemen.

De definitie van *oneindige verzameling* is nu simpel:

Definitie 4.2 *Verzameling A heet* **oneindig** *wanneer A niet eindig is.*

Het is gemakkelijk in te zien dat de verzameling van natuurlijke getallen $\mathbb{N}$ oneindig is volgens deze definitie. Ook de verzameling van alle zinnen van het Nederlands is oneindig. Datzelfde geldt voor de verzameling van alle zinnen van het Nederlands die met 'Ik' beginnen.

Opgave 4.1 Wat is er mis met de volgende redenering: "Als je een willekeurige Nederlandse bijzin 'B' neemt, en je zet daar 'dat Jan dacht dat' voor, dan krijg je een nieuwe Nederlandse (bij)zin. Hieraan kun je zien dat er Nederlandse zinnen zijn die je met behulp van dit recept altijd weer langer kunt maken. Daaruit volgt dat er in het Nederlands oneindig lange zinnen bestaan."

Als twee *eindige* verzamelingen even groot zijn, kunnen we dat ook uitleggen in termen van bi-jecties. Twee eindige verzamelingen A en B zijn *even groot* wanneer er een getal n is zo dat er een bi-jectie is van $\{x \in \mathbb{N} \mid x < n\}$ naar A en een bi-jectie van $\{x \in \mathbb{N} \mid x < n\}$ naar B. Maar dan is er ook een directe bi-jectie van A naar B (ga na).

Is het nu zo dat alle oneindige verzamelingen even groot zijn? Het ligt voor de hand om de bi-jectie-uitleg van 'even groot zijn' ook op oneindige verzamelingen te gaan toepassen. Daarbij stappen we—om de associatie met *eindigheid* te vermijden—over op een andere terminologie.

In plaats van over *even groot als* zullen we het nu hebben over *gelijkmachtig met*. Gelijkmachtigheid is een begrip dat zowel op eindige als op oneindige verzamelingen van toepassing is. Hier is de definitie van *gelijkmachtigheid* (Eng.: equipollence):

Definitie 4.3 *Verzameling A heet* **gelijkmachtig met** *verzameling B (notatie: $A =_1 B$) wanneer er een bi-jectie van A naar B bestaat.*

De definities van 'eindig' en 'oneindig' waren nogal prozaïsch. Het begrip *gelijkmachtigheid* maakt echter een meer opwindende blik op het oneindige mogelijk. De definitie van 'gelijkmachtigheid' heeft namelijk als merkwaardig gevolg dat bij voorbeeld de verzameling $\mathbb{N}$ en de verzameling $\mathbb{N} - \{1\}$ gelijkmachtig ('even groot') zijn. Immers, de functie $f : \mathbb{N} \to \mathbb{N} - \{1\}$ gedefinieerd door $f(n) = n + 1$ is een bi-jectie (zie het plaatje).

$$\begin{array}{ccc} 1 & \longrightarrow & 2 \\ 2 & \longrightarrow & 3 \\ 3 & \longrightarrow & 4 \\ 4 & \longrightarrow & 5 \\ 5 & \longrightarrow & 6 \\ & \vdots & \end{array}$$

We zien aan dit voorbeeld dat de oneindige verzameling $\mathbb{N}$ een echte deelverzameling heeft van dezelfde machtigheid. Het kan worden bewezen dat *elke* oneindige verzameling echte deelverzamelingen heeft van dezelfde machtigheid.

Aan de andere kant zal het je niet lukken een eindige verzameling te vinden die gelijkmachtig is met een van zijn echte deelverzamelingen. Dat dit geen toeval is blijkt uit de volgende stelling.

Stelling 4.1 *Als A een eindige verzameling is, dan is er geen echte deelverzameling van A waarmee A gelijkmachtig is.*

Bewijs: We gebruiken inductie naar het aantal elementen van A.

– Basisstap. A heeft nul elementen. In dit geval is A gelijk aan $\emptyset$, en $\emptyset$ heeft geen echte deelverzamelingen. De stelling is dus waar voor het basisgeval.

– Inductiestap. De inductie-hypothese luidt: als A hoogstens n elementen heeft, dan is er geen echte deelverzameling van A waarmee A gelijkmachtig is. Veronderstel dat A een verzameling is met $n + 1$ elementen. We doen een *reductio ad absurdum* (Sec. 1.6, blz. 11).

  Stel dat B een echte deelverzameling van A is waarmee A gelijkmachtig is. Dit wil zeggen: $B \subset A$, er is minstens één element a van A dat niet in B zit, en er is een bi-jectie f van A naar B. Noem het element van B waarop a door f wordt afgebeeld b. Beschouw nu de functie van B naar B die ontstaat door het domein van de functie f te beperken tot B. De gebruikelijke notatie hiervoor is $f \restriction B$. De functie $f \restriction B$ is een injectie van B naar B. Het bereik van deze functie is de verzameling $f[B] = \{f(x) \mid x \in B\}$. Deze bereik-verzameling is uiteraard een deelverzameling van B, maar het kan geen *echte* deelverzameling zijn van B. Dat zou in strijd zijn met de inductie-hypothese, die op B van toepassing is omdat B hoogstens n elementen heeft. Dus $f[B] = B$. Hieruit volgt dat $b = f(a')$ voor zeker element a' van B. Omdat f een bi-jectie is moet a' gelijk zijn aan a; dus: $a \in B$. Hiermee zijn we in tegenspraak geraakt met de veronderstelling die we over a hadden gemaakt. □

Opmerking tussendoor: hier en in het vervolg zullen we een bewijs steeds afsluiten met het symbool □.

Omdat elke oneindige verzameling gelijkmachtig is met echte deelverzamelingen van zichzelf, terwijl geen enkele eindige verzameling dat is, kunnen we—als we dat willen—*oneindige verzameling* definiëren als een verzameling die gelijkmachtig is met een van zijn echte deelverzamelingen.

Opgave 4.2 In het bewijs van Stelling 4.1 moet natuurlijk ergens gebruik worden gemaakt van de voorwaarde dat A eindig is, anders was die voorwaarde niet nodig.

Ga na op welke plek in het bewijs van Stelling 4.1 de eindigheid van A gebruikt wordt.

4.2 Aftelbaar versus overaftelbaar

De definitie van *gelijkmachtigheid* uit § 4.1 had als merkwaardig gevolg dat de verzameling $\mathbb{N}$ en de verzameling $\mathbb{N} - \{1\}$ gelijkmachtig ('even groot') zijn.

Het kan nog gekker: de verzameling van alle natuurlijke getallen $\mathbb{N}$ is 'even groot' als de verzameling $\mathcal{O}$ van de oneven natuurlijke getallen. Immers, $f : \mathbb{N} \to \mathcal{O}$, gedefinieerd door $f(n) = 2n + 1$, is een bi-jectie. Zie het plaatje:

$$\begin{array}{ccc} 1 & \longrightarrow & 1 \\ 2 & \longrightarrow & 3 \\ 3 & \longrightarrow & 5 \\ 4 & \longrightarrow & 7 \\ 5 & \longrightarrow & 9 \\ & \vdots & \end{array}$$

Net zo zijn er bi-jecties te vinden tussen de verzameling van natuurlijke getallen en de verzameling van even natuurlijke getallen, tussen de verzameling van even natuurlijke getallen en de verzameling van oneven natuurlijke getallen, tussen de verzameling van natuurlijke getallen en de verzameling van priemgetallen (getallen die alleen deelbaar zijn door zichzelf en door 1), enzovoorts.

Alle oneindige verzamelingen die we tot nu toe gezien hebben waren gelijkmachtig met $\mathbb{N}$. Voor dit soort oneindigheid voeren we een apart begrip in:

Definitie 4.4 *Een verzameling heet* **aftelbaar oneindig***(Eng.: denumerable, countably infinite) als hij gelijkmachtig is met* $\mathbb{N}$.

Definitie 4.5 *Een verzameling heet* **aftelbaar** *als deze eindig is, of aftelbaar oneindig.*

Elke eindige verzameling is dus aftelbaar, en sommige oneindige verzamelingen zijn dat ook.

Definitie 4.6 *Wanneer A aftelbaar oneindig is en f is een bi-jectie tussen A en $\mathbb{N}$, dan noemen we f een* **aftelling** *van A.*

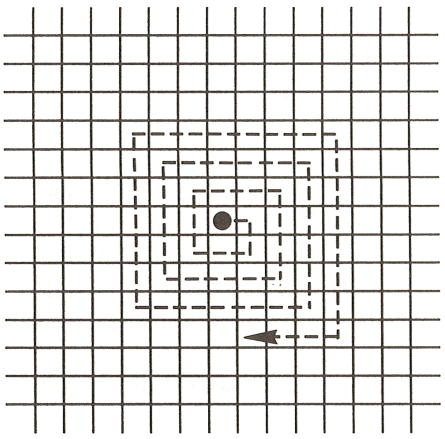
We geven nog een aantal voorbeelden van verzamelingen die gelijkmachtig zijn met $\mathbb{N}$.

Voorbeeld 4.4 Dat de verzameling van de gehele getallen, $\mathbb{Z}$, gelijkmachtig is met de verzameling ban de natuurlijke getallen, $\mathbb{N}$, volgt uit het bestaan van de volgende aftelling:

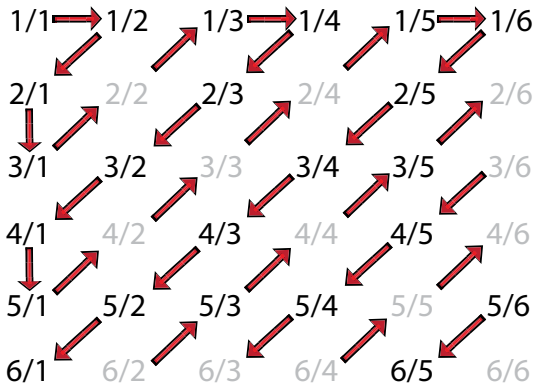
$$\begin{array}{ccc} 1 & \longrightarrow & 0 \\ 2 & \longrightarrow & 1 \\ 3 & \longrightarrow & -1 \\ 4 & \longrightarrow & 2 \\ 5 & \longrightarrow & -2 \\ & \vdots & \end{array}$$

Opgave 4.3 Geef een formele definitie van de bi-jectie uit voorbeeld 4.4.

Voorbeeld 4.5 Is de verzameling van alle velden van een oneindig schaakbord aftelbaar? Ja, kijk maar:



Voorbeeld 4.6 Moeilijker: is de verzameling van de positieve breuken aftelbaar? Het lijkt op het eerste gezicht van niet: tussen elk tweetal natuurlijke getallen liggen immers oneindig veel breuken. Cantor toonde echter aan dat de positieve breuken aftelbaar zijn, door de volgende fraaie aftelinstructie te geven:



Dit is nog niet helemaal een bi-jectie, want bepaalde getallen komen meerdere keren voor, telkens in een andere gedaante, bij voorbeeld 1/1, 2/2, 3/3, enzovoorts. Sla ze na de eerste keer gewoon over, en je hebt een bi-jectie.

Opgave 4.4 Laat zien dat de verzameling van **alle** breuken (positieve breuken, negatieve breuken, en het getal 0) aftelbaar is.

Voorbeeld 4.7 De verzameling van alle eindige rijtjes letters uit het alfabet is aftelbaar. Immers, deze verzameling kan als volgt worden gerangschikt: eerst de rijtjes van lengte 1 in alfabetische volgorde (dat zijn dus gewoon de letters van het alfabet: $a, b, c, \ldots$), dan de rijtjes van lengte twee in alfabetische volgorde ($aa, ab, ac, \ldots$), dan de rijtjes van lengte drie in alfabetische volgorde, enzovoort. Deze rangschikking is een aftelling. Merk op dat de grootte van dit (eindige) alfabet er niet toe doet: met 1000 in plaats van 26 letters kunnen we dezelfde redenering gebruiken. Zelfs kunnen we dit resultaat uitbreiden tot de verzameling rijtjes over een *aftelbaar* 'alfabet' A: zie opdracht 4.5.

Opgave 4.5 Bewijs dat de verzameling van alle eindige rijtjes symbolen uit een aftelbaar alfabet weer aftelbaar is. (Waarschuwing: je moet nu een andere aftelling maken dan in voorbeeld 4.7 hierboven).

Opgave 4.6 Ergens op een mooi plekje staat een uitzonderlijk groot hotel, een hotel met aftelbaar veel kamers: het Hilbert Hotel. Het hotel is genoemd naar de Duitse logicus en wiskundige David Hilbert. Op zekere dag zijn alle kamers bezet; er zijn dus aftelbaar veel gasten ondergebracht. Dan meldt zich nog iemand bij de receptie. De manager krabt zich even achter het oor, en bedenkt dan een manier om deze extra gast ook nog onder te brengen. Wat moet er gebeuren? (Hint: oneindig veel gasten die al zijn ondergebracht moeten verhuizen.)

Opgave 4.7 We zijn nog steeds bij het Hilbert Hotel, dat helemaal is volgeboekt. Er komt een bus voorrijden; geen gewone bus maar een Hilbert bus: er zitten aftelbaar veel passagiers in. Ook die worden allemaal ondergebracht. Hoe gebeurt dat?

Opgave 4.8 Juist als de portier van het Hilbert Hotel de deur op het nachtslot wil doen (er liggen aftelbaar veel gasten te ronken in aftelbaar veel kamers) komen aftelbaar veel Hilbert bussen voorrijden (elk met ... juist ja). IJlings wordt de manager gewekt. Valt hier nog iets aan te doen? Na enig heen en weer gepraat blijkt dat het Hilbert Hotel groot genoeg is om ook al deze gasten nog onder te brengen. Wat moet er gebeuren?

4.3 Ordening tussen verzamelingen

Tot nu toe hebben we gelijkmachtigheid van verzamelingen steeds aangetoond door het geven van een bi-jectief verband tussen die verzamelingen. In een aantal gevallen is het evenwel helemaal niet zo makkelijk die bi-jectie expliciet te geven. We volstonden bijvoorbeeld met het globaal aanduiden van een aftelling van $\mathbb{N}\times\mathbb{N}$ (of equivalent: de verzameling positieve breuken); het is waarachtig niet eenvoudig deze slingerende aftelling in een formule te omschrijven! Toch kunnen we dit soort resultaten zonder wiskundige hoogstandjes bewijzen. We moeten dan echter met *injecties* werken in plaats van met bi-jecties.

Voor eindige verzamelingen geldt dat als A kleiner is dan B, er een injectie van A naar B bestaat, en ook omgekeerd, als er zo'n injectie is, dan is A hoogstens even groot als B. De volgende definitie ligt nu voor de hand:

Definitie 4.7 *We zeggen dat verzameling A* **kleinere of gelijke machtigheid heeft** *dan verzameling B, en schrijven*

$$A \leq_1 B$$

wanneer er een injectie bestaat van A naar B.

Verder:

Definitie 4.8 *We zeggen dat verzameling A een* **strict kleinere machtigheid** *bezit dan verzameling B, en schrijven*

$$A <_1 B$$

wanneer $A \leq_1 B$ maar $A \neq_1 B$.

(Het symbool $=_1$ is gedefinieerd geworden in Definitie 4.3 op blz. 44.)

Opgave 4.9 Ga na dat

$$A \leq_1 B \;\Leftrightarrow\; A <_1 B \text{ of } A =_1 B.$$

Opgave 4.10 Bewijs:

1. $\mathbb{N} \leq_1 \mathbb{Q}$. (Hint: inbedding, Def. 3.26, blz. 41.)
2. $\mathbb{Q} \leq_1 \mathbb{R}$.
3. $[0,1] \leq_1 \mathbb{R}$.
4. Als $A \subseteq B$, dan $A \leq_1 B$.

Opgave 4.11 Bewijs:

1. $[0,2] \leq_1 [0,1]$.
2. $[0,10^6] \leq_1 [0,1]$.
3. $\mathbb{R}^+ \leq_1 [0,1]$. (Hint: bekijk functies als $x \mapsto x/(1+x)$ o.i.d.)
4. $\mathbb{R} \leq_1 [-1,1]$. (Gebruik dezelfde aanwijzing als in het vorige onderdeel.)

Opgave 4.12 Bewijs het volgende:

1. $\mathbb{N}\times\mathbb{N} =_1 \mathbb{N}$.
2. $\mathbb{Z}\times\mathbb{Z} =_1 \mathbb{Z}$.
3. $\mathbb{Q}\times\mathbb{Q} =_1 \mathbb{Q}$.
4. $\mathbb{R}\times\mathbb{R} =_1 \mathbb{R}$. (Het platte vlak bezit evenveel punten als de reële rechte.)

Hint voor de eerste drie onderdelen: bewijs eerst

$$A =_1 B \;\Rightarrow\; A\times A =_1 B\times B.$$

Hint voor het laatste onderdeel: bewijs eerst $(0,1)^2 =_1 (0,1)$, door coördinaten in het vierkant $(0,1)^2$ decimaal te representeren, en te bewijzen dat

$$(0.d_1d_2d_3\ldots, 0.d'_1d'_2d'_3\ldots) \mapsto 0.d_1d'_1d_2d'_2d_3d'_3\ldots$$

injectief is. Maak vervolgens gebruik van het feit $(0,1) =_1 \mathbb{R}$ (zie eerdere opgave).

Opmerking: in 1924 bewees A. Tarski dat $A =_1 A\times A$ voor *elke* oneindige verzameling, zelfs voor verzamelingen met een machtigheid groter dan $\mathbb{R}$ (zie "Tarski's theorem," blz. 52). In dit bewijs maakte Tarski gebruik van het keuzeaxioma (blz. 51). Sterker nog, hij toonde aan dat zijn stelling $A =_1 A\times A$ gelijkwaardig is aan het keuzeaxioma.

Opgave 4.13 Bewijs dat $\mathbb{R}^3 =_1 (0,1)$. (De ruimte bezit evenveel punten als het open eenheidsinterval.) Hint: bewijs eerst

$$A =_1 B \;\Rightarrow\; A\times C =_1 B\times C.$$

Gebruik vervolgens resultaten uit de vorige opgaven.

Opgave 4.14 1. Bewijs dat $\leq_1$ transitief is, i.e.,

$$A \leq_1 B \text{ en } B \leq_1 C \;\Rightarrow\; A \leq_1 C.$$

2. Bewijs: $A <_1 B$ en $B \leq_1 C \Rightarrow A <_1 C$ en $A \leq_1 B$ en $B <_1 C \Rightarrow A <_1 C$.

4.4 De stelling van Schröder-Bernstein

Na al het bovenstaande lijkt de volgende identiteit een niemendalletje en, analoog aan de vorige opgaven, makkelijk te bewijzen:

$$\text{als } A \leq_1 B \text{ en } B \leq_1 A, \text{ dan } A =_1 B.$$

In woorden: als er een injectie bestaat van A naar B en omgekeerd, dan zal er wel een bi-jectie bestaan tussen A en B.

Het zal blijken dat deze identiteit inderdaad waar is. Ook zal echter blijken dat deze identiteit lastig is te bewijzen. We gaan het echter wel doen! Deze plausibele maar lastig te bewijzen stelling staat bekend als de stelling van *Schröder-Bernstein*. Deze stelling staat ook bekend als de stelling van Cantor-Bernstein, of de stelling van Cantor-Schröder-Bernstein. Deze veelheid aan namen wordt veroorzaakt door het volgende. Cantor beschreef de stelling voor het eerst 1883, maar gaf er geen bewijs bij. Of hij ook een bewijs had was niet duidelijk. Wel gaf hij aan het bewijs in één van zijn volgende papers te publiceren. Cantor hield geen woord, althans, na meer dan tien jaar kwamen anderen met een bewijs. Ernst Schröder publiceerde een bewijs in 1996. Vermoedelijk onafhankelijk daarvan publiceerde Felix Bernstein een bewijs in 1997. Wat voor bewijs Cantor dan wel niet in gedachten had is helaas nooit duidelijk geworden...

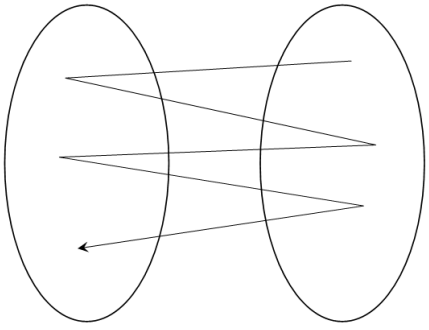
Stelling 4.2 (Schröder-Bernstein)

$$\textit{als } A \leq_1 B \textit{ en } B \leq_1 A, \textit{ dan } A =_1 B.$$

In woorden: als er een injectie bestaat van A naar B, en er bestaat een injectie van B naar A, dan bestaat er een bi-jectie van A naar B.

Zoals gezegd: de stelling van Schröder-Bernstein lijkt een trivialiteit, maar laat zich niet makkelijk bewijzen. Het volgende bewijs is nog één van de meest makkelijke.[1]

Bewijs: Laat $f : A \to B$ en $g : B \to A$ injectief. Om deze reden geldt voor beide functies dat het volledig origineel van één element altijd weer één element is. Hierdoor is het mogelijk zogenaamde zig-zag's te trekken. Een *zig-zag* wordt als volgt gemaakt.



Neem een element uit A of B en loop *terug* naar het volledig origineel (wat dus één punt is). Loop vanaf daar weer terug, en dan weer, en dan weer, enz. In sommige gevallen kun je niet verder. Dit eindpunt wordt dan de *bron* of *oorsprong* van alle punten in de zig-zag genoemd. Laat bijvoorbeeld $a_1 \in A$ en stel dat de zig-zag die begint in a_1 stopt in a_3:

$$.a_3 \to b_2 \to a_2 \to b_1 \to a_1$$

Dus $g^{-1}(a_3) = \emptyset$. Dan is a_3 de bron van zichzelf, de bron van b_2, de bron van a_2, van b_1, en van a_1. Nogmaals, sommige zig-zag's hebben geen oorsprong en gaan oneindig ver terug.

Vorm nu drie verzamelingen: A_A, A_B, A_∞:

- A_A: alle punten in A met oorsprong in A.
- A_B: alle punten in A met oorsprong in B.
- A_∞: alle punten in A zonder oorsprong.

Ga na dat deze drie verzamelingen een partitie (opdeling) van A vormen. Analoog kan B worden onderverdeeld in drie verzamelingen B_A, B_B, en B_∞.

Het is nu mogelijk A_A bi-jectief af te beelden op B_A, A_B bi-jectief af te beelden op B_B, en A_∞ bi-jectief af te beelden op B_∞, bijvoorbeeld door:

$$\begin{aligned} f|_{A_A}: &\quad A_A \to B_A \\ g|_{B_B}^{-1}: &\quad A_B \to B_B \\ f|_{A_\infty}: &\quad A_\infty \to B_\infty \end{aligned}$$

$f|_{A_A}$ is gewoon f, maar dan beperkt tot A_A.[2] Verder is $g|_{B_B}$ gelijk aan g beperkt tot B_B. Tenslotte is $g|_{B_B}^{-1}$ dan de inverse van $g|_{B_B}$, eventjes aangenomen dat deze bestaat en welgedefinieerd is. Plak deze drie bi-jecties aan elkaar, en je hebt je bi-jectie van A naar B.

We moeten nog wel eventjes nagaan dat de drie restricties ook echt bi-jecties zijn. Van $f|_{A_A}$ en $f|_{A_\infty}$ is dat makkelijk in te zien. (Check: welgedefinieerd, injectief en surjectief.) Dit zijn dus volwaardige bi-jecties. Bovendien blijven dan inderdaad de gebieden A_B en B_B over. Die zijn dan voor $g|_{B_B}$. Weer is makkelijk in te zien dat $g|_{B_B}$ bi-jectief is. Dus $g|_{B_B}^{-1}$ is welgedefinieerd en bi-jectief. □

Opgave 4.15 Zie je waarom in dit bewijs gekozen is voor $g|_{B_B}^{-1}$ en niet voor $(g^{-1})|_{A_B}$?

Voorbeeld 4.8 Een eenvoudige toepassing van de stelling van Schröder-Bernstein zien we bij de aftelbaarheid van $\mathbb{Q}$. Omdat $\mathbb{N} \subseteq \mathbb{Q}$, is er uiteraard een injectie van $\mathbb{N}$ naar $\mathbb{Q}$: de identieke functie f zodat $f(n) = n$ voor elke n. En voor de injectie g de andere kant op nemen we voor elke uitgedeelde breuk $\frac{m}{n}$ met $m, n \in \mathbb{N}$ de waarde

$$g(\frac{m}{n}) = 2^m 5^n$$

[1] Een vorige versie van deze tekst bevatte een bewijs waarbij "randjes" "omgevouwen" moesten worden. Dit bewijs is vervangen door het huidige bewijs, dat geïnspireerd is op bewijzen uit [Halmos 1960] en [Shen & Vereshchagin 2002].

[2] Zie blz. 39: "Uitbreidingen en restricties".

en voor negatieve breuken

$$g(-\frac{m}{n}) = 3^m 5^n.$$

Controleren dat dit injecties zijn (denk aan uniciteit priemontbinding, p. 13) en ... klaar.

Opmerking: een ander evident lijkend resultaat is dat $\leq_1$ een totale ordening is, dat wil zeggen dat elk tweetal verzamelingen A en B *altijd* in kardinaliteit kunnen worden vergeleken:

$$A =_1 B \text{ of } A <_1 B \text{ of } A <_1 B.$$

(*Trichotomie*-wet, of *wet van de uitgesloten derde*.) Dit is geen vanzelfsprekendheid en volgt uit het keuzeaxioma (blz. 51). Sterker nog, totaliteit van kardinaliteit is gelijkwaardig aan het keuzeaxioma.

Iets zwakkere (maar voor dit dictaat zeer bruikbare!) resultaten kunnen wel makkelijk worden bewezen. Deze resultaten berusten op het zogenaamde *duiventil-principe*.

Met de stelling van Schröder-Bernstein kan nu ook een alternatieve karakterisering van kleinere of gelijke machtigheid worden gegeven. Als voorbereiding daarvan eerst een opgave.

Opgave 4.16 Ga na dat $A <_1 B$ als en slechts als er een injectie bestaat van A naar B, maar niet omgekeerd.

Samengevat:

- $A \leq_1 B \Leftrightarrow$ er bestaat een injectie van A naar B.
- $A =_1 B \Leftrightarrow$ er bestaat een injectie van A naar B, en omgekeerd.
- $A <_1 B \Leftrightarrow$ er bestaat een injectie van A naar B, maar niet omgekeerd.

Met de stelling van Schröder-Bernstein mogen deze drie items dus nu ook gerust als definitie van kleinere of gelijke machtigheid worden gebruikt.

Het duiventil-principe (Eng.: pigeon hole principle) is een zeer fundamenteel principe in de informatica. Het zegt: als er n duiven in m hokken zitten, en $m < n$, dan is er tenminste één hok aan te wijzen waar twee of meer duiven in zitten. Het duiventil-principe gaat ook op voor oneindige verzamelingen:

Stelling 4.3 *Als $f : \mathbb{N} \to A$ een surjectie is, dan is A hoogstens aftelbaar.*

Bewijs: Er zijn twee mogelijkheden: ofwel de verzameling

$$B = \{n \in \mathbb{N} \mid \text{ er is geen } k \text{ in } \mathbb{N}, k < n, \text{ met } f(k) = f(n)\}$$

is eindig, of die verzameling is aftelbaar. In het eerste geval zitten er eindig veel elementen in A, in het tweede geval zijn dat er aftelbaar veel. Immers:

$$g = \{\langle n, f(n)\rangle \mid n \in B\}$$

is een bi-jectie van B naar A. □

Dit kunnen we veralgemeniseren. Dat gebeurd in de volgende twee opgaven.

Opgave 4.17 (Duiventil-principe) Laat zien dat als $f : A \to B$ en $B <_1 A$, dat f dan geen injectie kan zijn. (Hint: gebruik de definitie van $<_1$ en de stelling van Schröder-Bernstein.)

Opgave 4.18 *i)* Bewijs: $f : A \to B$ is surjectief $\Rightarrow$ $B \leq_1 A$. (Hint: volg de definitie van $\leq_1$ en gebruik het resultaat van Opgave 3.24, blz. 39.)

ii) Laat met behulp van het vorige onderdeel zien dat als $f : A \to B$ en $A <_1 B$, dat f dan geen surjectie kan zijn.

Weer merken we op dat er impliciet gebruik is gemaakt van het keuzeaxioma, in dit geval bij de beantwoording van onderdeel *i)*. Kun je aangeven op welke plek in je antwoord dat gebeurd?

Opgave 4.19 Zij A, B aftelbaar. Bewijs:

1. Als B eindig is, dan is $A \cup B$ aftelbaar.
2. $A \cup B$ is aftelbaar.
3. $A \cap B$ is aftelbaar.
4. $A \times B$ is aftelbaar.
5. $A \setminus B$ is aftelbaar.

De volgende opgave is een voortzetting van Opgave 4.19.

Opgave 4.20 Zij $A_1, \ldots, A_n$ aftelbaar. Bewijs:

1. $\bigcup_{i=1}^n A_i = A_1 \cup \cdots \cup A_n$ is aftelbaar.
2. $\bigcap_{i=1}^n A_i = A_1 \cap \cdots \cap A_n$ is aftelbaar.
3. $\prod_{i=1}^n A_i = A_1 \times \cdots \times A_n$ is aftelbaar.
4. Alle eindige en niet-complementaire verzamelingstheoretische combinaties van $A_1, \ldots, A_n$ zijn aftelbaar, dat wil zeggen: alle eindige verzamelingstheoretische combinaties van $A_1, \ldots, A_n$ zijn aftelbaar, zolang er gecombineerd wordt met operatoren uit $\{\cup, \cap, \times\}$.

Opgave 4.21 Laat aan de hand van een tegenvoorbeeld zien dat het volgende niet geldt: $A \subseteq X$ en A aftelbaar $\Rightarrow$ $X \setminus A$ aftelbaar. Maak je tegenvoorbeeld zo simpel mogelijk. Vergelijk het indien mogelijk met het tegenvoorbeeld van iemand anders.

Opgave 4.22 Zij $A_1, \ldots, A_n, \ldots$ een aftelbare rij van eindige verzamelingen. Bewijs:

1. $\bigcup_{i=1}^\infty A_i = A_1 \cup \cdots \cup A_n \cup \ldots$ is aftelbaar.
2. $\bigcap_{i=1}^\infty A_i = A_1 \cap \cdots \cap A_n \cap \ldots$ is aftelbaar.

In de beantwoording van de vorige opgave heb je naar alle waarschijnlijkheid onbewust gebruik gemaakt van het keuzeaxioma. Hier komen we nog over te spreken op blz. 51.

Opgave 4.23 Zij $A_1,\ldots,A_n,\ldots$ een aftelbare rij van aftelbare (mogelijk oneindige) verzamelingen. Bewijs:

1. $\bigcup_{i=1}^{\infty} A_i = A_1 \cup \cdots \cup A_n \cup \ldots$ is aftelbaar.
2. $\bigcap_{i=1}^{\infty} A_i = A_1 \cap \cdots \cap A_n \cap \ldots$ is aftelbaar.

Het begint er een beetje op te lijken dat alle oneindige verzamelingen gelijkmachtig zijn met $\mathbb{N}$. Dat dat *niet* zo is, is door Cantor aangetoond. Cantor liet zien dat de machtsverzameling van $\mathbb{N}$ niet gelijkmachtig is met $\mathbb{N}$.

Allereerst herinneren we eraan dat in § 3.5 in essentie al bewezen is dat $2^{\mathbb{N}} =_1 \{0,1\}^{\mathbb{N}}$; er was immers een 1–1-correspondentie tussen deelverzamelingen en karakteristieke functies. We redeneren daarom verder met $\{0,1\}^{\mathbb{N}}$ in plaats van $2^{\mathbb{N}}$. Dat $\{0,1\}^{\mathbb{N}}$ *minstens* even groot is als $\mathbb{N}$ is gemakkelijk in te zien (zie volgende opdracht).

Opgave 4.24 Laat zien dat $\{0,1\}^{\mathbb{N}}$ minstens even groot is als $\mathbb{N}$. Je kunt dit doen door het aangeven van een *injectie* van $\mathbb{N}$ naar $\{0,1\}^{\mathbb{N}}$.

Cantor bewees dat $\{0,1\}^{\mathbb{N}}$ wezenlijk groter is dan $\mathbb{N}$, met behulp van zijn beroemde diagonaal-argument.

Stelling 4.4 (Diagonaalstelling)

De verzameling $\{0,1\}^{\mathbb{N}}$ is niet aftelbaar.

Bewijs: Stel dat de verzameling $\{0,1\}^{\mathbb{N}}$ wél aftelbaar zou zijn. Dan zouden we een aftelling $f_0, f_1, f_2, f_3, \ldots$ hebben van karakteristieke functies. We leiden een tegenspraak af door het construeren van een karakteristieke functie f^* die *niet* in de aftelling voorkomt.

De functies $f_0, f_1, f_2, \ldots$ uit de aftelling waarvan we het bestaan veronderstellen kunnen als volgt worden gerangschikt (dit is natuurlijk maar een voorbeeld; de functies zouden er ook anders kunnen uitzien):

	0	1	2	3	4	5	6	…
f_0	**1**	0	0	0	0	0	0	…
f_1	0	**1**	0	1	0	0	1	…
f_2	1	0	**0**	1	1	0	0	…
f_3	0	0	0	**0**	1	1	0	…
f_4	1	0	0	0	**0**	1	1	…
f_5	1	0	0	0	0	**1**	0	…
f_6	1	0	0	0	0	0	**1**	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Beschouw nu de waarheidswaarde-toekenningen op de oneindige diagonaal van dit plaatje. De functie f^* die je krijgt door deze waarheidswaarden *om te keren* (in het voorbeeld: de functie f^* met $f^*(0)=0, f^*(1)=0, f^*(2)=1, f^*(3)=1, f^*(4)=1, f^*(5)=0$, enzovoort) is *ongelijk* aan elke functie in de aftelling. Immers, hij verschilt van f_i in de waarde die aan het i-de argument wordt toegekend. Hiermee is de veronderstelling dat $\{0,1\}^{\mathbb{N}}$ kan worden afgeteld weerlegd. □

Het is hopelijk duidelijk waarom de redeneerwijze uit deze stelling 'Cantor's diagonaal-argument' wordt genoemd. Het diagonaal-argument kan ook worden gebruikt om te laten zien dat de verzameling van de reële getallen (gebruikelijke aanduiding: $\mathbb{R}$) niet aftelbaar is. De reële getallen zijn alle breuken plus alle irrationale getallen. Als je weet dat zo'n reëel getal kan worden geschreven als een decimale breuk die achter de komma oneindig doorloopt, kun je zelf bedenken hoe het diagonaal-argument hier moet worden gehanteerd. Dat $\mathbb{R}$ niet aftelbaar is, is overigens geen toeval: $\{0,1\}^{\mathbb{N}}$ is gelijkmachtig met de verzameling $\mathbb{R}$ der reële getallen (zie [Van Dalen e.a. 1975] voor het bewijs).

Cantor's diagonaalstelling levert ons een eerste voorbeeld van een verzameling die oneindig is, maar niet aftelbaar. Hiervoor voeren we nieuw jargon in:

Definitie 4.9 *Een oneindige verzameling die niet aftelbaar is heet* **overaftelbaar** *(Eng.: non-denumerable, uncountable).*

Opgave 4.25 Laat op dezelfde manier als boven zien dat de verzameling van alle functies van $\mathbb{N}$ naar $\mathbb{N}$, i.e, de verzameling

$$\mathbb{N}^{\mathbb{N}},$$

overaftelbaar is.

Opgave 4.26 Laat zien dat de verzameling van alle eindige deelverzamelingen van $\mathbb{N}$ daarentegen wél aftelbaar is.

Opgave 4.27 Een **co-finiete** deelverzameling van $\mathbb{N}$ is een deelverzameling A van $\mathbb{N}$ met de eigenschap dat $\mathbb{N} - A$ eindig is (dus: een verzameling met een eindig complement). Laat zien dat de verzameling van alle *co-finiete* deelverzamelingen van $\mathbb{N}$ aftelbaar is.

Cantor's diagonaalstelling is in feite een speciaal geval van een algemenere stelling, die ook door Cantor is bewezen. Deze stelling voert ons via steeds hogere machtigheden binnen in een wiskundig "paradijs" van steeds grotere oneindige verzamelingen.[3]

Stelling 4.5 (Algemene diagonaalstelling) *Eke verzameling A bezit een strict kleinere machtigheid dan haar machtsverzameling 2^A:*

$$A <_1 2^A$$

[3] In een artikel getiteld "Über das Unendliche" in de Mathematische Annalen 95 (1926) schreef David Hilbert: *"Aus dem Paradies, das Cantor uns geschaffen, soll uns niemand vertreiben können."* Dit schreef hij omdat er diverse figuren, waaronder met name Bertrand Russell, zaten te knagen aan Cantor's verzamelingenleer. Hilbert hoopte dat de fundamenten van de wiskunde hierdoor niet zouden worden aangetast. Meer hierover in Sectie 4.8 op blz. 56.

Bewijs: Voor het gemak werken we eerst het geval af dat $A = \emptyset$. In dit geval is $|A| = |\emptyset| = 0$ en $|2^A| = |\{\emptyset\}| = 1$, dus de stelling geldt.

Voor niet lege A laten we twee dingen zien. In de eerste plaats: 2^A heeft minstens dezelfde machtigheid als A. In de tweede plaats: A en 2^A hebben niet dezelfde machtigheid.

Het eerste is gemakkelijk: de functie $f : A \to 2^A$ die wordt gedefinieerd door $f(a) = \{a\}$, voor elke $a \in A$, is een injectie. Het tweede gaat weer volgens de strategie van de vorige stelling. We nemen aan dat een bi-jectie F tussen A en 2^A gegeven is, en we construeren een deelverzameling B van A die geen F-beeld is van enig element in A. De constructie van $B \subseteq A$ gaat als volgt. Kies

$$B = \{b \in A \mid b \notin F(b)\}.$$

B is geen F-beeld van enig element van A. Veronderstel namelijk van wel. Neem aan dat we hebben: $F(c) = B$, voor zekere $c \in A$. Zit c in B? Stel van wel. Dan volgt uit de definitie van B dat $c \notin B$. Stel van niet. Dan volgt uit de definitie van B dat $c \in F(c)$, dat wil zeggen $c \in B$. Beide mogelijkheden leiden dus tot een tegenspraak. Uit het feit dat B geen F-beeld is van enig element van A volgt dat we mogen concluderen dat er geen bi-jectie tussen A en 2^A bestaat. □

De volgende opgave is een vervolg op Opgave 4.23. Je dient deze opgave dus eerst gemaakt te hebben.

Opgave 4.28 Zij $A_1, \ldots, A_n, \ldots$ een aftelbare rij van eindige verzamelingen.

1. Bewijs dat
$$\prod_{i=1}^{\infty} A_i = A_1 \times \cdots \times A_n \times \ldots$$
in het algemeen niet aftelbaar is, zelfs niet als alle A_i eindig zijn.

2. Onder welke voorwaarden is het oneindige Cartesisch product wél aftelbaar? Als je een antwoord gevonden hebt, probeer je voorwaarden dan wat op te rekken [door (sommige?) A_i groter te maken]. Hoe ver kun je daar in gaan?

Opgave 4.29 1. Zij $\{A_i\}_{i=1}^{\infty}$ een aftelbare oneindige rij van aftelbaar oneindige deelverzamelingen van de vorm

$$A_i = \{0,1,0,0,0,1,1,0,0,0,1,1,1,1,1,0,0,\ldots\}$$

voor $i \geq 1$. Bewijs, door de A_i onder elkaar te zetten in een oneindige "tabel," dat $\bigcup_{i=1}^{\infty} A_i$ aftelbaar is.

2. Bewijs, voorzover je dat nog niet hebt gezien of gedaan, met behulp van een oneindige tabel dat de collectie van alle naar rechts oneindige deelrijen uit $\{0,1\}$, i.e., rijen van de vorm

$$0,1,0,0,0,1,1,0,0,0,1,1,1,1,1,0,0,\ldots$$

overaftelbaar oneindig is.

3. Leg uit waarom de tabellen en resultaten in 1 en 2 niet met elkaar in tegenspraak zijn.

Opgave 4.30 Een veelterm (ook wel: polynoom) over $\mathbb{Q}$ is een uitdrukking van de vorm

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,$$

met $n \geq 0$, en coëfficiënten $a_i \in \mathbb{Q}$, bijvoorbeeld:

$$\frac{1}{5}x^8 - \frac{6}{11}x^2 + \frac{3}{7}x - \frac{4}{9} = 0.$$

De verzameling van veeltermen over $\mathbb{Q}$ wordt genoteerd als $\mathbb{Q}[X]$ (inderdaad: hoofdletter X). Bewijs het volgende, met gebruikmaking van de resultaten van bovenstaande opgaven:

1. Elementen uit $\mathbb{Q}[X]$ kunnen, zonder verlies of toevoegingen van oplossingen (nulpunten), omgezet worden in elementen uit $\mathbb{N}[X]$.

2. De verzameling $\mathbb{N}[X]$ is aftelbaar.

3. De verzameling reële nulpunten van veeltermen over $\mathbb{N}$ is aftelbaar.

4. De verzameling reële nulpunten van veeltermen over $\mathbb{Q}$ is aftelbaar. (Er zijn tenminste twee bewijzen mogelijk.)

5. De verzameling *algebraïsche getallen*, $\mathbb{A}$ is als volgt gedefinieerd:

   $$\mathbb{A} = \{x \in \mathbb{R} \mid x \text{ is nulpunt van een veelterm over } \mathbb{Q}\}.$$

   Bewijs dat $\sqrt{2}$ en $\sqrt[3]{5/8}$ algebraïsch zijn.

6. Bewijs dat $\mathbb{A}$ aftelbaar is. (Dit kan kort, onder verwijzing naar één van de eerdere onderdelen.)

7. Een reëel getal heet *transcendent* (vrij vertaald: "overstijgend") als het geen algebraïsch getal is. De verzameling van transcendente getallen kan genoteerd worden met $\mathbb{T}$. Dit is geen algemeen geldende notatie.

   Bewijs dat er reële transcendente getallen bestaan.

8. Bewijs dat er meer elementen in $\mathbb{T}$ dan in $\mathbb{A}$ zitten.

Er kan worden bewezen dat getallen zoals e en π transcendent zijn. Dit valt echter buiten het bestek van deze tekst.

Opgave 4.31 Een *Diofantische vergelijking* is de nulstelling van een veelterm in meerdere variabelen, waarbij de coëfficiënten geheeltallig zijn, bijvoorbeeld

$$3xy^2 - 7x^3z + 5xyz^4 - 8 = 0.$$

Bewijs dat de verzameling van Diofantische vergelijkingen aftelbaar is. (Hint: probeer te werken in strata (lagen). Een stratum is dan een verzameling Diofantische vergelijkingen met een vaste limiet op het aantal termen, variabelen, factoren in termen, hoogte van exponenten, enz.)

4.5 Bijna alle

De kreet “bijna alle” heeft een speciale betekenis in de wiskunde en de informatica. Voor aftelbare (dus eindige en aftelbaar oneindige) verzamelingen betekent het: “alle op eindig veel na”. Wordt een overaftelbare verzameling als referentie-set (“achtergrond”) gebruikt, dan betekent het: “alle op aftelbaar veel na”.

Opgave 4.32 Welke van de volgende beweringen zijn waar? (Misschien is het handig niet eerder bij de antwoorden kijken voordat alles is beantwoord.)

1. Bijna iedereen op de campus heeft paars haar.
2. Bijna alle mensen op deze wereld zijn geboren op Mars.
3. Bijna alle zandkorrels op deze wereld zijn roze.
4. Bijna alle materie in het heelal is door kabouter Plop geschapen.
5. Bijna alle priemgetallen zijn oneven.
6. Bijna alle priemgetallen zijn deelbaar door 7.
7. Bijna alle elementen van $\mathbb{R}$ zijn niet geheel.
8. Bijna alle elementen van $\mathbb{R}$ zijn geen breuk.
9. Bijna alle elementen van $\mathbb{R}$ zijn niet algebraïsch (zijn geen oplossing van een geheeltallige veelterm).
10. Bijna alle oneindige bitstrings hebben oneindig veel 1-en.

Bijna altijd

De kreet “bijna altijd” betekent: “met kans 1”. Dit is niet gelijk aan “altijd”. Denk aan het herhaaldelijk gooien met een dobbelsteen: bijna altijd gooi je op den duur een zes. Maar niet altijd. Het is kans-theoretisch mogelijk dat het je met een zuivere dobbelsteen nooit lukt een zes te gooien. Kansrekening valt buiten dit dictaat en we gaan hier niet verder op in.

“Bijna alle” en “bijna altijd” hebben met elkaar te maken. Ze gaan allebei over het meten van verzamelingen. Als bijna alle B niet A zijn, dan zegt men officieel dat A *Lebesgue maat* nul heeft in B. Als bijna nooit A plaatsvindt als B plaatsvindt, dan zegt men officieel dat A *kansmaat* nul heeft in B. Een kansmaat is een bijzonder geval van een Lebesgue maat.

4.6 Het keuzeaxioma

In de beantwoording van verschillende opgaven (waaronder Opgave 4.22 op blz. 48 en Opgave 3.24 op blz. 39) heb je (naar alle waarschijnlijkheid onbewust) gebruik gemaakt van het keuzeaxioma (Eng.: *axiom of choice*, AC).

Definitie 4.10 (Keuzeaxioma) *Voor elke collectie van niet-lege verzamelingen bestaat een functie die uit elke verzameling één element kiest.*

Formeel: voor elke collectie van niet-lege verzamelingen

$$\mathscr{S} = \{S_i \mid i \in I\}$$

bestaat er een functie $f : \mathscr{S} \to \cup\mathscr{S}$ *zó dat* $f[S] \in S$.

Voorbeeld: bekijk voor elk vak dat dit jaar gegeven wordt aan een Nederlandse universiteit de inschrijvinglijst. Laat $\mathscr{S}$ de collectie van deze inschrijvinglijsten zijn. Een keuzefunctie kiest uit elke lijst een cursist. Dat is alles. Opmerkingen:

1. Zo’n functie wordt een keuzefunctie genoemd.
2. Elk gekozen element heet een *representant*.
3. Elementen uit $\mathscr{S}$ hoeven niet disjunct te zijn (dezelfde persoon kan meerder vakken volgen). Eenzelfde element (persoon) kan dus representant zijn van meerdere verzamelingen (vakken).
4. $\mathscr{S}$ mag leeg zijn, maar haar elementen niet.

Het zojuist gegeven voorbeeld betrof een eindige collectie van verzamelingen. Nu is bekend dat bij *eindige* gevallen het keuzeaxioma kan worden gemist. (we kunnen dan immers gewoon concreet, i.e. echt, kiezen.) Pas bij oneindige collecties van verzamelingen zal het keuzeaxioma het verschil gaan maken. We geven dus nog een voorbeeld.

Uit elke denkbare regio van het aardoppervlak (dat zijn er oneindig veel als we het aardoppervlak idealiseren als het oppervlak van een bol) valt een punt te selecteren. Uiteraard kan zoiets op verschillende manieren, maar het keuzeaxioma zegt nu juist dat het *überhaupt* kan. We hebben expres voor een boloppervlak gekozen omdat elementen in regio’s zich minder duidelijk laten ordenen dan elementen van, bijvoorbeeld, intervallen in $\mathbb{R}$. (Uiteindelijk kan het wel, en die laatste conclusie dat zoiets kan voor alle verzamelingen is weer equivalent aan het keuzeaxioma!)

Om beter inzicht te krijgen voor het keuzeaxioma is het misschien handig de volgende parabel te lezen. Deze parabel is afkomstig van de Britse filosoof Bertrand Russell (1919). Ga er maar eens even lekker voor zitten en steek desnoods een pijp op.

De gepensioneerde vrijgezelle miljonair

Ergens in Engeland leeft een gepensioneerde vrijgezelle miljonair. Deze is zó rijk dat hij een garderobe bezit met aftelbaar oneindig veel sokken en aftelbaar oneindig veel schoenen! Elk tweetal schoenen is duidelijk van elkaar te onderscheiden, op een manier die voor alle schoenen kan worden toegepast: van elk paar schoenen is er een linker- en een rechterschoen. Daarentegen zijn in elk paar de sokken niet van elkaar te onderscheiden.

De miljonair verveelt zich en vraagt zijn butler van elk paar schoenen er één te brengen, waarbij het niet van de butler mag afhangen met welke schoenen deze terug komt; de butler mag zelf niet kiezen. Daarom vraagt hij: "welke schoenen wilt u dat ik pak"? De miljonair antwoordt: "breng alle rechterschoenen maar". Vanaf dat moment hoeft de butler niets meer te vragen en weet hij welke schoen van elk paar gebracht moet worden.

Op een gegeven moment vraagt de miljonair zijn butler van elk paar sokken er één te brengen. Weer mag het niet van de butler afhangen met welke sok van elk paar hij terug komt. De butler vraagt: "welke sokken wilt u dat ik pak"? De butler mag niet zelf kiezen, dus de miljonair mag de butler niet instrueren zelf maar wat te pakken.

Voor een eindige kast met sokken is de opdracht triviaal. De miljonair kan, als hij zich kwaad maakt, naar de kast toelopen en paar voor paar aanwijzen welke sok hij wil hebben, waarna de butler de opdracht alsnog kan uitvoeren. (Vrij zinloos, maar het kan.) Voor een bak met aftelbaar oneindig veel sokken werkt dit NIET.

Wiskundigen en filosofen hebben lang nagedacht over dit schijnbaar (!) triviale probleem. In 1963 bewees de Amerikaanse wiskundige Paul Cohen (1934-2007) dat de taal van standaard verzamelingenleer een dergelijk keuzevoorschrift niet kan genereren. Cohen bewees dus dat het onmogelijk is de butler een opdracht mee te geven om sokken te selecteren zonder dat deze zelf keuzes moet gaan maken. Meer bepaald liet Cohen zien dat, om te bewijzen dat een aftelbare collectie van paren (zonder generiek onderscheidingscriterium) zelf ook weer aftelbaar is (Opgave 4.22!), we moeten aannemen dat uit elk paar één element gekozen kan worden. Dit heet *het axioma van de aftelbare keuze*, notatie AC_ω. Een sterker axioma, *het keuzeaxioma*, notatie AC, stipuleert dat bij een *willekeurige* collectie van niet-lege verzamelingen (aftelbaar of niet) uit elke verzameling een representant gekozen kan worden.

Veel stellingen in de wiskunde kunnen niet worden bewezen zonder gebruikmaking het keuzeaxioma. De volgende resultaten zijn zelfs *gelijkwaardig* aan het keuzeaxioma:

– Het Cartesisch product van iedere niet-lege collectie van niet-lege verzamelingen is niet leeg.

– (Tarski's theorem.) Voor elke oneindige verzameling A bestaat een bi-jectie tussen A en $A \times A$. (Zie Opgave 4.12, blz. 46.)

– (Trichotomie.) Kardinaliteit-ordening tussen verzamelingen is *totaal*: voor elk tweetal verzamelingen, A en B, geldt: $A =_1 B$, $A <_1 B$ of, $B <_1 A$. (Vergelijk de stelling van Schröder-Bernstein.)

– Iedere surjectieve functie bezit een rechts-inverse.

– Elke verzameling bezit een wel-ordening.

– (Zorn's lemma.) Elke partiëel geordende verzameling waarin elke keten (totaal geordende set) naar boven begrensd is, bezit een maximaal element (element dat niet kleiner is dan andere elementen).

Ontzettend veel meer stellingen zijn gelijkwaardig aan AC, en nog veel meer stellingen steunen op het keuzeaxioma, maar hier we laten het bij. Merk op dat sommige van deze stellingen al zijn besproken zonder dat we ons druk hebben gemaakt of AC gold. Dit is typisch voor AC: je gebruikt het, vaak zonder het zelf te weten.

Als je ons tot hier toe hebt gevolgd dan zou je misschien kunnen zeggen: "als het keuzeaxioma klaarblijkelijk zo nodig is, waarom accepteren we het dan niet en gaan we gewoon door? Waarom doen jullie zo moeilijk over het keuzeaxioma?" Daar zijn twee redenen voor.

Ten eerste is het keuzeaxioma het enige axioma in de verzamelingenleer dat niet constructief is: het poneert het bestaan van een functie zonder die functie zelf te geven. Dat is onbevredigend. Kijk bijvoorbeeld nog eens naar het bewijs van de stelling dat er getallen $a, b \notin \mathbb{Q}$ bestaan, zó dat $a^b \in \mathbb{Q}$ (blz. 14). Dit bewijs is niet constructief: je komt te weten dat getallen $a, b \notin \mathbb{Q}$ bestaan, zó dat $a^b \in \mathbb{Q}$, zónder dat de concrete getallen je werkelijk worden aangereikt. Voor sommigen wiskundigen, genaamd *constructivisten*, is dat onacceptabel.

Ten tweede blijkt dat, als we er voor kiezen het keuzeaxioma te accepteren, er niet alleen stellingen kunnen worden bewezen waarvan we vinden dat die waar moeten zijn, maar óók dat er stellingen kunnen worden bewezen waarvan we vinden dat die merkwaardig of zelfs tegen-intuïtief zijn. De volgende resultaten volgen namelijk *ook* als we besluiten het keuzeaxioma te accepteren:

- Er bestaan gebeurtenissen in de kansrekening waarvan de kans niet kan worden bepaald, hoewel de gebeurtenis op zichzelf exact genoeg is omschreven.

  Een voorbeeld van een dergelijke gebeurtenis is de volgende: laten we zeggen dat twee punten op de eenheidscirkel tot dezelfde *familie* behoren als en slechts als we (over de cirkel) van het ene naar het andere punt kunnen lopen met passen ter lengte 1. (Ga na dat er verschillende families bestaan.) Het keuzeaxioma zegt nu dat voor elke familie een representant kan worden gekozen. Er kan nu worden bewezen dat de kans om een representant te treffen bij aselect kiezen van een punt op de eenheidscirkel, per definitie onbepaald is!?

- (Banach-Tarski paradox.) Het is mogelijk een massieve bol zó te versnijden dat de stukken kunnen worden samengevoegd tot twee bollen met hetzelfde volume (Fig. 4.1)!? De versnijding kan plaats vinden in eindig veel stukken en de assemblage kan plaatsvinden middels standaard translaties en rotaties. (Hier volgt trouwens niet uit dat we uit één bol van één kilogram twee bollen van één kilogram kunnen maken. Het volume van de versneden stukken is niet meetbaar. Hierin schuilt ook de "truc" van de Banach-Tarski paradox.)

- (Winnende strategieën.) Jij en Bob gaan een spel spelen:

Figuur 4.1: De Banach-Tarski paradox (tekening B.D. Esham, in Inkscape).

– Bob denkt aan een willekeurige functie $f : \mathbb{R} \to \mathbb{R}$. (Kan van alles zijn, hoeft bv. niet continu te zijn.)
– Jij kiest een $x_0 \in \mathbb{R}$.
– Bob geeft je alle functiewaarden, behalve die op x_0.
– Jij probeert nu $f(x_0)$ te voorspellen.

Jij wint als je voorspelling klopt, anders verlies je. Onder aanname van AC blijkt dat er een strategie bestaat waarmee je dit spel wint met kans 1!?

De gebruikelijke verzamelingenleer, waarop alle wiskunde wordt voortgebouwd, wordt aangeduid met ZFC: axioma's van Zermelo & Fraenkel + het keuzeaxioma. Er is ook een tak van wiskunde die voortbouwt op ZF zonder AC. Dit wordt *constructieve wiskunde* genoemd. Sommige vanzelfsprekend lijkende resultaten over oneindige verzamelingen kunnen dus niet meer bewezen worden in de constructieve wiskunde.

4.7 Equivalentieklassen en kardinaalgetallen

We zullen nu gaan uitleggen hoe de relatie *gelijkmachtig zijn met* kan worden gebruikt om zogenaamde *kardinaalgetallen* te definiëren: getallen die de grootte van (eindige of oneindige) verzamelingen aangeven.

Wanneer we de klasse V van alle verzamelingen beschouwen, dan is $=_1$ een relatie op V. Deze relatie heeft de volgende eigenschappen:

– $=_1$ is reflexief. Immers: voor elke verzameling A geldt dat er een bi-jectie van A naar A bestaat; de identieke functie op A, $\{\langle a,a\rangle \mid a \in A\}$, die we noteren als id_A, is zo'n bi-jectie.
– $=_1$ is symmetrisch. Immers: als f een bi-jectie is tussen A en B, dan is f^{-1} een bi-jectie tussen B en A. Hieruit volgt meteen: als $A =_1 B$, dan $B =_1 A$.
– $=_1$ is transitief. Immers: als f een bi-jectie is van A naar B, en g is een bi-jectie van B naar C, dan is de compositiefunctie $g \circ f$ een bi-jectie van A naar C (ga dit zelf na; teken een plaatje). Dus: als $A =_1 B$ en $B =_1 C$, dan $A =_1 C$.

Opgave 4.33 Welke eigenschappen heeft de relatie $\leq_1$?

Om wat naders te kunnen zeggen over de eigenschappen van de relatie $=_1$ stappen we even over naar eigenschappen van relaties in het algemeen.

Definitie 4.11 *Een tweeplaatsige relatie R op een verzameling A die de drie eigenschappen reflexiviteit, symmetrie en transitiviteit bezit heet een* **equivalentierelatie**.

Voorbeeld 4.9 Op de verzameling M van mensen is de relatie *even groot zijn als* een equivalentierelatie.

Dat dit zo is is gemakkelijk na te gaan:

– Ieder mens is even groot als hijzelf;
– als persoon a even groot is als persoon b, dan is persoon b even groot als persoon a;
– als persoon a even groot is als persoon b, en persoon b is even groot als persoon c, dan is persoon a even groot als persoon c.

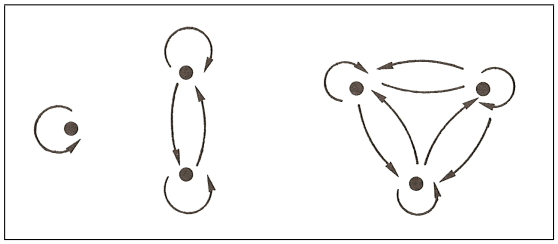
Voorbeeld 4.10 Op de verzameling M van mensen is de relatie *even oud zijn als* een equivalentierelatie. (Ga dit zelf na.)

Voorbeeld 4.11 Op de verzameling $\mathbb{N}$ van de natuurlijke getallen is de relatie R gedefinieerd door xRy desda $x + y$ is even een equivalentierelatie.

We gaan even na dat dit een equivalentierelatie is:

– voor elk getal n geldt dat $2n$ even is, dus R is reflexief;
– als de som van m en n even is, dan is de som van n en m dat ook, dus R is symmetrisch;
– als de som van m en n even is, en de som van n en k ook, dan zijn m en k hetzij allebei even, hetzij allebei oneven, dus dan is de som van m en k ook even. Dus: R is transitief.

Voorbeeld 4.12 De pijl-relatie in het volgende plaatje is een equivalentierelatie (ga zelf na).



Opgave 4.34 Bedenk zelf nog twee voorbeelden van equivalentierelaties. Neem als domein: de verzameling van alle mensen.

Gegeven het feit dat R een equivalentierelatie is op een verzameling A kunnen we gaan kijken naar de *equivalentieklasse* van een element a van A.

Definitie 4.12 *Zij A een verzameling, en R een equivalentierelatie op A. De* **equivalentieklasse** *van een element a van A, modulo R, is de verzameling van alle elementen van A die in de R-relatie staan tot a.*

Opgave 4.35 1. Hoeveel equivalentierelaties zijn er mogelijk op $\{1,2,3\}$? (Hint: denk in klassen.)

2. Bewijs met volledige inductie dat het aantal mogelijke equivalentierelaties op n elementen gelijk is aan B_n, waarbij
$$B_n = \sum_{j=0}^{n-1} \binom{n-1}{k} B_k.$$
B staat voor het *Bell-getal*.

   Hint 1: bekijk de klassen van een willekeurige equivalentierelatie op n elementen, voor het gemak op $\{1,\ldots,n\}$. Het element n moet ergens een plek krijgen:
$$\{\{\ldots\},\ldots\{\ldots\},\{\ldots,n\}\}$$
   (volgorde maakt niet uit). Tel het aantal klasgenoten, j, van n. Er blijven nog $n-1-j$ elementen over om klassen van te maken:
$$\{\underbrace{\{\ldots\},\ldots\{\ldots\}}_{n-1-j},\underbrace{\{\ldots}_{j},n\}\}$$

   Hint 2: het aantal mogelijke klasgenoten van n loopt van 0 tot $n-1$.

   Hint 3: het aantal mogelijk equivalentierelaties op $n-1-j$ elementen is per definitie B_{n-1-j}.

Notatie voor de equivalentieklasse van $a \in A$, modulo R: $[a]_R$. Elk element van de equivalentieklasse $[a]_R$ heet een **representant** van $[a]_R$. Nog wat jargon: we zeggen dat de equivalentierelatie R op A een verzameling van equivalentieklassen **induceert**. Dit jargon geeft aanleiding tot een nieuwe definitie.

Definitie 4.13 *De* **quotiëntverzameling** *van A modulo R is de verzameling equivalentieklassen die op A geïnduceerd wordt door een equivalentierelatie R.*

Notatie voor de quotiëntverzameling van A modulo R: A/R. Merk op dat A/R de verzameling $\{[a]_R \mid a \in A\}$ is.

Voorbeeld 4.13 De deelverzameling van de verzameling van alle mensen die gegeven is door $[\text{Jan van Eijck}]_{\text{Even-oud-als}}$ is de verzameling van de mensen die even oud zijn als Jan van Eijck. Jan van Eijck is een representant van die verzameling.

Opgave 4.36 De relatie R op $\mathbb{N}$ gedefinieerd door xRy desda $x+y$ is even, induceert twee equivalentieklassen. Ga na welke deelverzamelingen van $\mathbb{N}$ dit zijn.

De relatie *even oud zijn als* induceert (op een gegeven moment) ruim honderd equivalentieklassen op de verzameling van alle mensen (namelijk de klasse van mensen die nog geen jaar oud zijn, de klasse van mensen die een jaar oud zijn, de klasse van mensen die twee jaar oud zijn, enzovoorts tot we bij de oudste aardbewoner zijn aangekomen.

Opgave 4.37 Hoeveel equivalentieklassen zijn er dan als de oudste persoon 120 jaar is?

Opgave 4.38 Hoeveel verschillende equivalentieklassen induceert de pijlrelatie op de verzameling uit voorbeeld 4.12?

Strikt genomen moeten we nog een paar dingen *bewijzen* over de verzameling van equivalentieklassen die wordt geïnduceerd door een equivalentierelatie.

Stelling 4.6 *Als A een verzameling is en R een equivalentierelatie op A, dan is elk element a van A lid van een element van de verzameling van equivalentieklassen die geïnduceerd wordt door R.*

Bewijs: Zij A een verzameling, en zij R een equivalentierelatie op A. Omdat R reflexief is, hebben we voor elk element a van A: aRa. Hieruit volgt meteen:
$$a \in \{y \in A \mid yRa\},$$
dat wil zeggen $a \in [a]_R$. Dus: elke $a \in A$ is lid van *minstens* één equivalentieklasse. □

Dat elke $a \in A$ lid is van *hoogstens* één equivalentieklasse volgt uit de volgende stelling:

Stelling 4.7 *Als A een verzameling is, en R is een equivalentierelatie op A, dan geldt, voor willekeurige elementen a en b uit A: als $[a]_R \cap [b]_R \neq \emptyset$ dan $[a]_R = [b]_R$.*

Bewijs: Stel dat $[a]_R \cap [b]_R \neq \emptyset$. Met andere woorden: er is een $c \in A$ met cRa en cRb. Neem nu een willekeurig element x van $[a]_R$. We hebben dan xRa. Uit cRa en de symmetrie van R volgt: aRc. Transitiviteit van R levert nu: xRc. We hadden al cRb, dus nogmaals transitiviteit van R toepassen levert op: xRb. Met andere woorden: $x \in [b]_R$. Omdat x een willekeurig element van $[a]_R$ was, hebben we hiermee aangetoond dat $[a]_R \subseteq [b]_R$. Op dezelfde manier valt aan te tonen dat $[b]_R \subseteq [a]_R$. Dus $[a]_R = [b]_R$. □

Blijkbaar deelt de verzameling equivalentieklassen die geïnduceerd wordt door een equivalentierelatie R op A, de verzameling A op in *onderling disjuncte* deelverzamelingen (deelverzamelingen die onderling geen enkel element gemeenschappelijk hebben). Zo'n opdeling van een verzameling V heet een *partitie* van V. Iets preciezer gezegd:

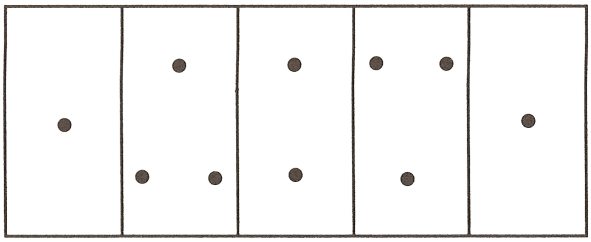
Definitie 4.14 *Een verzameling $\mathscr{A}$ van deelverzamelingen van A heet een* **partitie** *van A als $\mathscr{A}$ voldoet aan de volgende voorwaarden:*

– $A = \bigcup\{X \mid X \in \mathscr{A}\}$;

– *voor alle* $X, Y \in \mathscr{A}$: $X = Y$ *of* $X \cap Y = \emptyset$.

De eerste voorwaarde zegt dat elk element $a \in A$ lid is van *minstens* één verzameling uit de partitie; de tweede voorwaarde zegt dat elk element $a \in A$ lid is van *hoogstens* één verzameling uit de partitie.

We gaan weer over op aanschouwelijk onderricht. Hier is een plaatje van een partitie van een verzameling:



Opgave 4.39 Het is eenvoudig in te zien dat elke partitie $\mathscr{A}$ op een verzameling A ondubbelzinnig een equivalentierelatie vastlegt. Geef de equivalentierelatie R die de partitie in het bovenstaande plaatje induceert.

Dit was een uitweiding over equivalentierelaties. Terug naar ons verhaal over de relatie $=_1$ die we hierboven hebben ingevoerd om de grootte van verzamelingen te peilen. Uit het feit dat $=_1$ een equivalentierelatie is volgt dat $=_1$ een partitie induceert op de klasse van alle verzamelingen. Voor elke verzameling A kunnen we nu de equivalentieklasse van A modulo $=_1$ beschouwen, dat wil zeggen de klasse $[A]_{=_1} = \{B \mid B =_1 A\}$.

Twee verzamelingen X en Y zitten in dezelfde equivalentieklasse modulo $=_1$ wanneer er een bi-jectie bestaat van X naar Y. Dus: intuïtief gesproken zitten twee verzamelingen in dezelfde equivalentieklasse modulo $=_1$ als ze 'evenveel' elementen hebben; deze equivalentieklassen duiden dus de *grootte* van verzamelingen aan. Om deze reden noemden Cantor, Frege en Russell dergelijke equivalentieklassen *kardinaalgetallen*. Het kardinaalgetal 0 is volgens deze definitie niets anders dan de equivalentieklasse $[\emptyset]_{=_1}$. Het kardinaalgetal 1 is gelijk aan de equivalentieklasse $[\{\emptyset\}]_{=_1}$.

Opgave 4.40 Hoeveel elementen telt de equivalentieklasse $[\emptyset]_{=_1}$? Hoeveel elementen telt een representant van deze equivalentieklasse?

Opgave 4.41 Hoeveel elementen telt een representant van de equivalentieklasse $[2^{\{\emptyset\}}]_{=_1}$?

In plaats van de wat moeizame notatie $[V]_{=_1}$ schrijven we meestal $|V|$. Het kardinaalgetal dat de grootte aangeeft van de verzameling $\mathbb{N}$ duiden we aan als $\aleph_0$ (spreek uit: 'alef nul'; $\aleph$ is de eerste letter uit het Hebreeuwse alfabet); we schrijven dus $|\mathbb{N}| = \aleph_0$. We zeggen ook: "de *kardinaliteit* van een aftelbare verzameling is alef nul." Evenzo is er de afspraak voor de kardinaliteit voor de verzameling reële getallen dat $|\mathbb{R}| = \aleph$.

Als we willen *rekenen* met kardinaalgetallen zullen we moeten afspreken wat we daarbij bedoelen met optellen, vermenigvuldigen en machtsverheffen. Opnieuw geldt dat deze definities zijn geïnspireerd door het eindige geval:

Definitie 4.15 *Als voor de verzamelingen A en B geldt dat $A \cap B = \emptyset$, $|A| = k$ en $|B| = \ell$, dan zijn $k + \ell$, $k \cdot \ell$, k^ℓ , $k = \ell$, $k \leq \ell$ en $k < \ell$ gedefinieerd door:*

$$
\begin{aligned}
&k + \ell = |A \cup B| \\
&k \cdot \ell = |A \times B| \\
&k^\ell = |A^B| \\
&k = \ell \Leftrightarrow A =_1 B \\
&k \leq \ell \Leftrightarrow A \leq_1 B \\
&k < \ell \Leftrightarrow k \leq \ell \;\&\; k \neq \ell.
\end{aligned}
$$

Het feit dat A en B disjunct zijn speelt eigenlijk alleen bij de optellingsregel een rol; maar ook daar is die rol marginaal want *gegeven* k en ℓ kunnen we altijd disjuncte A en B kiezen van geschikte kardinaliteit. In alle gevallen moet nog wel bewezen worden dat de resulterende kardinaliteit (binnen de gestelde condities) niet afhangt van de keuze van A en B; we laten dit over aan de lezer.

Voor de rekenkundige bewerkingen op kardinaalgetallen gelden een aantal normale algebraïsche eigenschappen:

Stelling 4.8 *Voor willekeurige kardinaalgetallen k, ℓ en m geldt:*

$k + \ell = \ell + k$	*commutativiteit van* $+$
$k \cdot \ell = \ell \cdot k$	*commutativiteit van* $\cdot$
$k + (\ell + m) = (k + \ell) + m$	*associativiteit van* $+$
$k \cdot (\ell \cdot m) = (k \cdot \ell) \cdot m$	*associativiteit van* $\cdot$
$k \cdot (\ell + m) = k \cdot \ell + k \cdot m$	*distributiviteit*
$k + \ell \leq k \cdot \ell$.	

Bewijs: Deze gelijkheden volgen direct uit de bijbehorende eigenschappen van operaties op verzamelingen. Kies om de ongelijkheid aan te tonen disjuncte verzamelingen A en B zodat $|A| = k$ en $|B| = \ell$, en vaste $a \in A$ en $b \in B$, dan is er een injectie i van $A \cup B$ naar $A \times B$ zodat voor elke x: $i(x) = \langle x, b\rangle$ als $x \in A$ en $i(x) = \langle a, x\rangle$ als $x \in B$. □

Stelling 4.9 *Voor willekeurige kardinaalgetallen k, ℓ en m geldt:*

$$
\left.
\begin{aligned}
&k^\ell \cdot k^m = k^{\ell+m} \\
&k^\ell \cdot m^\ell = (k \cdot m)^\ell \\
&(k^\ell)^m = k^{\ell \cdot m}
\end{aligned}
\right\} \quad \textit{exponent–wetten}
$$

Bewijs: Nogal bewerkelijk, maar niet echt lastig. We laten dit over aan de lezer. □

De belangrijkste feiten over $\aleph_0$ en $\aleph$ zijn al genoemd (of zelfs bewezen) in § 4.2:

Stelling 4.10 *Er geldt:*

$$\aleph_0 + \aleph_0 = \aleph_0 \qquad \aleph_0 \cdot \aleph_0 = \aleph_0$$
$$\aleph_0 + \aleph = \aleph \qquad \aleph_0 \cdot \aleph = \aleph$$
$$\aleph + \aleph = \aleph \qquad \aleph \cdot \aleph = \aleph$$
$$\aleph_0 < 2^{\aleph_0} = \aleph = \aleph_0^{\aleph_0} = \aleph^{\aleph_0}.$$

Bewijs: zie § 4.2 voor een bewijs van de eerste regel en de vaststelling dat $2^{\aleph_0} = \aleph$. We maken daarna een sprong naar vermenigvuldiging van $\aleph$. Met gebruik van wat eerder bewezen is, volgt nu $\aleph \cdot \aleph = 2^{\aleph_0} \cdot 2^{\aleph_0} = 2^{\aleph_0 + \aleph_0} = 2^{\aleph_0} = \aleph$. We kunnen nu door *scherp afschatten* ($\leq$ is vanwege 'Schröder-Bernstein' anti-symmetrisch) hieruit de optellingseigenschap voor $\aleph$ afleiden:

$$\aleph \leq \aleph + \aleph \leq \aleph \cdot \aleph = \aleph.$$

De andere eigenschappen zijn analoog te bewijzen. □

In § 3.5 hebben we al laten zien dat er voor iedere verzameling A een bijectie bestaat tussen 2^A en $\{0,1\}^A$. De algemene diagonaalstelling toont dus aan dat het kardinaalgetal van $\{0,1\}^A$ groter is dan dat van A, voor elke A, met andere woorden, voor elk kardinaalgetal $k : k < 2^k$. Derhalve moet er een rij van steeds groter wordende oneindige kardinaalgetallen bestaan. Het kleinste oneindige kardinaalgetal is het kardinaalgetal van $\mathbb{N}$, dat we $\aleph_0$ hebben gedoopt. We noemen het eerstvolgende kardinaalgetal $\aleph_1$, het daaropvolgende $\aleph_2$, enzovoorts. Zo ontstaat de zogenaamde 'rij der alefs', de rij van oneindige kardinaalgetallen gerangschikt naar grootte:

$$\aleph_0, \aleph_1, \aleph_2, \aleph_3, \aleph_4, \ldots$$

We weten echter ook dat $\aleph_0 < \aleph$. Dus waar zit $\aleph$ in de rij van de $\aleph_n$? Deze vraag heeft de gemoederen van wiskundigen en logici gedurende vele decennia beziggehouden. Omdat $\aleph$ het kardinaalgetal is van $\mathbb{R}$, en $\mathbb{R}$ als een (continue) getallenlijn wordt voorgesteld, wordt dit probleem dat van de 'machtigheid van het continuum' genoemd. De discussie over deze vraag werd geopend met de presentatie van Cantor's 'transfinite Mengenlehre' (1895), en afgesloten met de publicatie van een artikel van Paul Cohen (1963).

Cantor vermoedde dat er geen verzamelingen bestaan met een machtigheid die tussen die van $\mathbb{N}$ en $\{0,1\}^{\mathbb{N}}$ in ligt. Hij wist al dat de machtigheid van $\mathbb{R}$ gelijk is aan die van $\{0,1\}^{\mathbb{N}}$. Zijn vermoeden luidde dus dat de machtigheid van het continuum het kardinaalgetal is dat onmiddellijk volgt op $\aleph_0$, dat wil zeggen: $\aleph_1$. Het lukte hem echter niet om dit te bewijzen. Cantor's *continuumhypothese* luidde als volgt:

Hypothese 4.1 (CH) $\aleph = \aleph_1$.

Omdat $2^{\aleph_0} = \aleph$, mogen we de continuumhypothese ook als volgt formuleren:

Hypothese 4.1 (CH–herformulering) $2^{\aleph_0} = \aleph_1$.

Deze formulering geeft aanleiding tot het opstellen van een *algemene continuumhypothese*, afgekort 'ACH' (Eng.: Generalized Continiuum Hypothesis, GCH) waarvan CH een speciaal geval is:

Hypothese 4.2 (ACH) *Voor alle* $\alpha : 2^{\aleph_\alpha} = \aleph_{\alpha+1}$.

Dat dit niet slechts een bewering over alefs is, blijkt beter in de equivalente formulering:

Hypothese 4.2 (ACH–herformulering) *Voor elke oneindige k geldt dat er geen kardinaalgetal tussen k en* 2^k *in ligt.*

In 1938 toonde Kurt Gödel aan dat, gesteld dat de zogenaamde Zermelo-Fraenkel axioma's voor de verzamelingenleer consistent zijn (zie ook § 4.8), het toevoegen van ACH deze theorie in elk geval niet inconsistent maakt. In 1963 liet Cohen zien dat als de Zermelo-Fraenkel axioma's consistent zijn, ook het toevoegen van de *negatie* van ACH de theorie niet inconsistent maakt. Kortom, we kunnen beide kanten uit: we kunnen zowel ACH als ¬ACH toevoegen aan ZF zonder het systeem in gevaar te brengen. We zeggen wel: ACH is *onafhankelijk* van de Zermelo-Fraenkel axioma's.

4.8 Paradoxen

Je hebt wellicht gemerkt dat we nu al een paar keer het woord *klasse* hebben gebruikt op plaatsen waar je misschien het woord *verzameling* zou hebben verwacht: we hebben gesproken over de *klasse* van alle verzamelingen en over de *klasse* $[A]_{=_1}$. Dit spraakgebruik is niet toevallig; het is bedoeld om enkele voetangels en klemmen van de naïeve verzamelingenleer te vermijden.

Cantor's formulering van de verzamelingenleer (nu aangeduid als *naïeve verzamelingenleer*) leidde tot het optreden van tegenspraken, in deze context meestal 'paradoxen' genoemd. De bekendste contradictie werd in 1902 door Bertrand Russell (1872–1970) gesignaleerd; zij staat bekend als de Russell-paradox. Voor een huis-, tuin- en keukenverzameling V geldt altijd $V \notin V$. Bij voorbeeld: de verzameling van alle theelepeltjes is zelf geen theelepeltje. Beschouw nu de verzameling $R = \{V \mid V \notin V\}$. Dan geldt dat R in zichzelf zit als hij de kenmerkende eigenschap bezit, maar dan heeft hij zichzelf juist niet tot element. In formule:

$$R \in R \Longleftrightarrow R \notin R.$$

En dit is absurd, zoals je al zonder veel logische training kunt vaststellen.

Cantor had overigens zelf drie jaar voor Russell al een probleem gevonden in zijn theorie van kardinaalgetallen, maar daar pas in 1932 over gepubliceerd. Veronderstel maar dat er een verzameling van alle verzamelingen bestaat, een *universele verzameling* U. Dan $2^U \subseteq U$ en dus $2^U \leq_1 U$, maar dit is in strijd met de algemene diagonaalstelling (stelling 4.5).

Ook de Russell-paradox volgt uit de aanname van het bestaan van een universele verzameling U. Merk op dat dan $U \in U$, en dat is al vreemd, maar het echte probleem ontstaat door uit U de deelverzameling R te lichten van alle verzamelingen die geen element zijn van zichzelf.

Om beide paradoxen te vermijden moeten we in elk geval af van de aanname dat er een verzameling van alle verzamelingen bestaat. Dat kan bij voorbeeld als volgt. We gaan uit van een aantal bona fide verzamelingen (met name: de lege verzameling en de verzameling van de natuurlijke getallen), en van een aantal bona fide operaties op verzamelingen (met name: verenigen, doorsnijden en het vormen van machtsverzamelingen), dat wil zeggen: het uitvoeren van deze operaties op bona fide verzamelingen levert weer bona fide verzamelingen op. Verder nemen we het zogenaamde *separatie-axioma* aan:

Axioma 4.1 (separatie) *Als je een verzameling X hebt en een eigenschap E, dan bestaat er een verzameling Y die precies die elementen uit X bevat met eigenschap E.*

Nu kunnen we de volgende stelling formuleren:

Stelling 4.11 *Uit elke verzameling A kunnen we met de 'Russell-eigenschap'* **geen element zijn van zichzelf** *een verzameling B lichten. We hebben dan:*
$B = \{V \mid V \in A \text{ en } V \notin V\} \notin A$.

Bewijs: Dat $B = \{V \mid V \in A \text{ en } V \notin V\}$ een verzameling is volgt uit het feit dat A een verzameling is plus het separatie-axioma. We moeten nu nog aantonen dat $B \notin A$. Uit de definitie van B volgt dat voor alle W:

$$W \in B \iff W \in A \text{ en } W \notin W.$$

Wat voor alle W geldt, geldt zeker ook voor B. Dus:

$$B \in B \iff B \in A \text{ en } B \notin B. \tag{1}$$

We passen nu een *reductio ad absurdum* toe. Neem aan dat we zouden weten dat $B \in A$. Daarmee zou (1) gereduceerd worden tot een contradictie:

$$B \in B \iff B \notin B.$$

Dus de aanname dat $B \in A$ is onjuist, en daarmee is bewezen dat $B \notin A$. □

Uit deze stelling volgt meteen dat er geen universele verzameling bestaat (anders gezegd: dat de *klasse* van alle verzamelingen zelf geen verzameling is).

We zijn nu toe aan de moraal van dit alles voor ons verhaal over gelijkmachtige verzamelingen. De moraal is deze: er is geen garantie dat de equivalentieklassen die door de relatie $=_1$ worden geïnduceerd op de klasse van alle verzamelingen zelf bona fide verzamelingen zijn. Sterker nog, het zijn *geen* bona fide verzamelingen, getuige de volgende stelling:

Stelling 4.12 *Er bestaat geen verzameling A zodanig dat A alle verzamelingen V bevat die precies één element hebben. Met andere woorden: de equivalentieklasse $[\{\emptyset\}]_{=_1}$ is geen verzameling.*

Bewijs: Stel dat de equivalentieklasse $[\{\emptyset\}]_{=_1}$ wel een verzameling is. Noem die verzameling A. We hebben nu:

$$V \in A \iff V =_1 \{\emptyset\}.$$

We hebben aangenomen dat het verenigen van verzamelingen een bona fide operatie is. Dus: als A een verzameling is, dan is de verzameling die ontstaat door de vereniging te nemen van alle elementen van A, dat wil zeggen de verzameling $\bigcup A$, ook een verzameling. De elementen van A zijn precies alle singletons. Dus voor elke verzameling V geldt dat $\{V\} \in A$. Hieruit volgt dat $\bigcup A$ elke verzameling bevat, ofwel: $\bigcup A$ is de verzameling van alle verzamelingen. Zo'n verzameling bestaat niet, dus de aanname dat A een verzameling is, is onjuist.
$A = [\{\emptyset\}]_{=_1}$ is geen verzameling. □

Dit is vervelend, want het betekent dat de truc om via het overstappen van verzamelingen op hun equivalentieklassen modulo $=_1$ te abstraheren van alle verschillen tussen verzamelingen behalve hun aantallen elementen, strikt genomen niet geoorloofd is. De manier van Cantor, Frege en Russell om het begrip *kardinaalgetal* in te voeren is te simpel. Kardinaalgetallen in deze zin bestaan niet.

In de axiomatische verzamelingenleer worden kardinaalgetallen via een omweg langs zogenaamde *ordinaalgetallen* ingevoerd. Het behandelen van deze definitie zou hier te ver voeren; we volstaan daarom met een verwijzing naar het leerboek [Van Dalen e.a. 1975]. Er is nog een andere, eenvoudiger manier om problemen met de Cantor-Frege-Russell definitie van *kardinaalgetal* te vermijden. We kunnen afspreken dat we ons bij het vormen van equivalentieklassen zullen beperken tot een verzameling die bona fide is en tevens 'groot genoeg'.

Neem bij voorbeeld aan dat we een verzameling $\mathscr{A}$ hebben waar de lege verzameling $\emptyset$ in zit, plus elk natuurlijk getal, plus de hele verzameling $\mathbb{N}$, plus alles wat je kunt krijgen door een eindig aantal malen de operaties *doorsnijden*, *verenigen* en *machtsverzamelingen vormen* toe te passen op elementen van $\mathscr{A}$. Zo'n verzameling is bona fide, en voor ons in eerste instantie groot genoeg. Zij bevat bijvoorbeeld $\mathbb{N}$, $2^{\mathbb{N}}$, $2^{2^{\mathbb{N}}}$, etc. als elementen.

De verzameling bevat ook functies met deze elementen als domein en als co-domein: functies zijn niets anders dan verzamelingen geordende paren, en geordende paren kunnen zelf weer als verzamelingen worden gecodeerd. Voor dit laatste hebben we een trucje nodig dat door de wiskundige Von Neumann is voorgesteld: we spreken af dat $\langle a,b\rangle$ (het geordende paar van a en b) een afkorting is voor $\{\{a\},\{a,b\}\}$.

Opgave 4.42 Laat zien dat geordende paren, als ze met behulp van deze definitie worden ingevoerd, hun essentiële eigenschap

$$\langle a,b\rangle = \langle c,d\rangle \iff a = c \text{ en } b = d$$

blijven houden.

Opgave 4.43 Bedenk een andere codering voor het geordende paar $\langle a, b \rangle$ en bewijs hiervoor de bovenstaande essentiële eigenschap.

Als we nu kardinaalgetallen definiëren als equivalentieklassen modulo $=_1$ op de verzameling $\mathscr{A}$ is het gevaar van genoemde contradicties op een eenvoudige manier bezworen. Dat we bij onze definitie de moederverzameling $\mathscr{A}$ gebruiken vermelden we nu eens en voor al; we zullen het in het vervolg niet steeds in herinnering roepen.

Dit alles neemt niet weg dat de verzamelingenleer, nog wel het fundament van de moderne wiskunde, een enorme 'bug' bleek te bevatten. Die is er nu uit verwijderd, maar wie weet zijn er andere tegenspraken die we nog niet ontdekt hebben. Het zou prettig zijn als we voor eens en altijd konden *bewijzen* dat de (ingeperkte) verzamelingenleer geheel vrij van contradicties, ofwel *consistent*, is. Om dit te onderzoeken moet de verzamelingenleer exacter gedefinieerd worden. Inderdaad hebben de verzamelingsparadoxen geleid tot logische preciseringen zoals de al genoemde Zermelo-Fraenkel axioma's. Om deze predikatenlogische formules en het onderzoek naar hun onderlinge consistentie te begrijpen is het zaak wat meer over logica te weten te komen.

Hoofdstuk 5

Berekenbaarheid

5.1 Inleiding

Berekenbaarheid is een fascinerend maar lastig onderwerp: je redeneert over wat programma's nog net wel (of net niet) kunnen, meestal zonder zelf echt te programmeren.

> *Computability theory has a fearsome reputation for incomprehensibility and difficulty, even amongst logicians. When, many years ago, I attended my first logic conference and was asked my area of study by a senior logician, I was a little disconcerted by his response of, "Good luck—you'll need it," to my reply of "computability theory".* "Dexter" op amazon.com, in een 2005 review van Barry Cooper's boek *Computability Theory* (2003).

Laten we er niet omheen draaien en meteen de twee meest belangrijke definities geven.

Definitie 5.1 (Beslisbaarheid) *Een deelverzameling X van $\mathbb{N}$ heet* beslisbaar *wanneer er een computerprogramma geschreven kan worden dat voor elk gegeven element x van $\mathbb{N}$ kan uitmaken of $x \in X$ dan wel $x \notin X$.*

Definitie 5.2 (Opsombaarheid) *Een deelverzameling X van $\mathbb{N}$ heet* opsombaar *wanneer er een computerprogramma geschreven kan worden die de elementen van X in één of andere volgorde opsomt.*

Over hoe dat opsommen precies werkt, en wat er kan en mag bij dat opsommen, komen we later te spreken. Ook is het mogelijk het begrip "computerprogramma" verder te preciseren, maar dat komt ook later aan de orde, om precies te zijn in Sectie 5.3.

Uiteraard zal een computerprogramma dat een oneindig grote verzameling X van $\mathbb{N}$ opsomt nooit stoppen; het zal tot in lengte van dagen getallen uitbraken. Elke $x \in X$ zal vroeg of laat (na verloop van eindig veel tijd) worden afgedrukt, al weten we natuurlijk niet wanneer.

Terug naar beslisbaarheid. Je vind het misschien vreemd, maar er bestaan deelverzamelingen van $\mathbb{N}$ die niet beslisbaar zijn. Dit wil zeggen: er zijn verzamelingen natuurlijke getallen die de eigenschap hebben dat er geen computerprogramma te schrijven valt dat bij invoer van een willekeurig natuurlijk getal uit kan maken of dat getal tot de verzameling behoort of niet. Nog erger: er zijn verzamelingen natuurlijke getallen waarvoor zelfs geen programma te schrijven valt dat de getallen uit de verzameling in een of andere volgorde *opsomt* (zo dat je, als je maar lang genoeg bij de computer blijft staan, elk getal uit die verzameling vroeg of laat te voorschijn ziet komen). Met andere woorden: er bestaan deelverzamelingen van $\mathbb{N}$ die niet opsombaar zijn.

Strikt genomen dienen beweringen over beslisbaarheid en opsombaarheid allereerst te worden gepreciseerd vooraleer ze kunnen worden bewezen. Dat gaan we dit hoofdstuk doen.

Een eerste poging om beslisbaarheid en opsombaarheid te preciseren werden gelijktijdig en (vermoedelijk) onafhankelijk gedaan door de wiskundigen Alonzo Church en Alan Turing. De geschiedenis wijst uit dat Church ietsje eerder was. Hij stelde voor de notie *beslissingsprocedure* te preciseren als *μ-recursieve procedure*, een begrip waarvoor hij een zeer precieze wiskundige definitie beschikbaar had. Alan Turing, aan de andere kant, had het idee om *beslissingsprocedure* te lezen als: 'procedure die, als hij wordt uitgevoerd op een Turingmachine, in eindig veel stappen tot een resultaat leidt.' Bij dat voorstel hoorde natuurlijk weer een precieze—wiskundige—definitie van wat een Turingmachine is.

Het preciseringsvoorstel voor *beslisbaar zijn* van Church wordt wel de these van Church genoemd. Die these luidt als volgt:

Hypothese 5.1 (These van Church) *Elke mechanische beslissingsprocedure is een μ-recursieve procedure.*

Turing formuleerde eenzelfde inzicht:

Hypothese 5.2 (These van Turing) *Elke mechanische beslissingsprocedure kan door een Turing-machine worden geïmplementeerd.*

Het werd al gauw duidelijk dat de beide preciseringsvoorstellen voor *beslissingsprocedure* op hetzelfde neerkwamen: de procedures die een Turingmachine in eindig veel stappen kan uitvoeren zijn precies de μ-recursieve procedures.

Een bewijs leveren van de Church-Turing these is niet mogelijk: het is immers een voorstel om het vage en intuïtieve begrip *mechanisch beslisbaar* te lezen als *μ-recursief*, of, *op een Turing-machine implementeerbaar*. De Church-Turing these is in feite geen echte hypothese maar veeleer een *uitdaging* om met een mechanische beslissingsprocedure (concreet: computerprogramma) aan te komen die niet μ-recursief is (in de precieze zin van de wiskundige definitie). Zie verder Sectie 5.8.

Tegenwoordig zijn nagenoeg alle programmeertalen Turing-compleet, dat wil zeggen: minstens zo krachtig als een Turing machine (niet zo verrassend), maar (wellicht

verrassender) ook niet krachtiger! Wél is het in de ene programmeertaal makkelijker programmeren dan in de andere, en dat maakt het verschil. Laten we vanaf nu dan maar afspreken dat, als we over algoritmen praten, we mogen denken aan programma's in onze favoriete programmeertaal J (bijvoorbeeld Java of C#), met dien verstande dat je ook nog mag aannemen dat je werkt op een supercomputer, dat wil zeggen,

- dat **tijd** (wel bestaat maar) niet echt een probleem is,
- dat je altijd over meer dan voldoende **geheugen** (RAM en page file) beschikt, en dat de gebruikte integers (incl. array indices en pointers) onbegrensd zijn.

In het redeneren over wat computers wel en niet kunnen nemen we dus aan dat programma's met onbeperkte tijd en middelen mogen worden uitgevoerd. Kortweg: we programmeren met onbeperkte reserves. (Eng.: *unlimited resources*.) **Let op:vrijv** "onbeperkt" betekent *niet* "oneindig"! Het betekent: in willekeurige lange, maar *eindige*, tijd, met gebruikmaking van willekeurig veel, maar *eindig* veel, geheugen. Het programma moet dus uitvoerbaar zijn met een begrensd geheugen en op den duur stoppen.

5.2 Gödel nummering

Het concept Gödel nummering zorgt er voor dat programma's kunnen worden genummerd, en dus aftelbaar zijn. Bovendien zorgt een Gödel nummering (er zijn er meerdere) er voor dat de aftelling kan worden uitgevoerd door een computerprogramma. Een Gödel nummering zorgt er dus voor dat een *aftelling* tevens een *opsomming* is.

We identificeren J voor het gemak met alle (syntactisch correcte) programma's in de taal J. Dus $j \in J \Leftrightarrow j$ is een (syntactisch correct) programma in de taal J. Als programma j ariteit k heeft (k input-waarden nodig heeft), dan noteren we dat met j/k.

Het volgende resultaat zegt dat alle programma's in J achter elkaar kunnen worden gezet. We zullen het vaak nodig hebben.

Stelling 5.1 *De verzameling J is aftelbaar.*

Bewijs:vrijv Iedere programmeertaal J (bijv. Java) is opgebouwd uit een alphabet A (bv. ASCII). Nu geldt dat elk J-programma een string is, opgebouwd uit elementen van A, i.e, $J \subseteq A^+$, waarbij A^+ gelijk is aan alle niet-lege strings opgebouwd uit A, i.e.,

$$A^+ = \{ax \mid a \in A, x \in \{A^+, \epsilon\}\},$$

waarbij ϵ de lege string is. Het is dus voldoende te bewijzen dat A^+ aftelbaar is. Laat $A^n \subseteq A^+$ alle strings zijn ter lengte n. Dan is A^n eindig en

$$A^+ = \cup_{n=1}^{\infty} A^n.$$

Om A af te tellen sommen we eerst alle elementen van A^1 op: `a, b, c, d,...`. Dat zijn er eindig veel, dus op enig moment zijn we klaar met het opsommen van A_1. Dan alle elementen van A^2: `aa, ab, ac, ad,..., ba, bb, bc, bd,...`, dan alle elementen van A^3: `aaa, aab, aac, aad,...`, dan alle elementen van A^4, enzovoort. Als programma $j \in J$ bijvoorbeeld n karakters lang is, dan zal j verschijnen in de aftelling van A^n. □

Opgave 5.1 1. Stel dat er op dit moment N verschillende programmeertalen zijn: $J_1, \ldots, J_N$, neem N maar lekker groot. Beschouw de verzameling van alle programma's die met deze talen te maken zijn. Laat zien dat deze verzameling ook aftelbaar is. (Je kunt een direct bewijs geven of minzaam verwijzen naar één van de resultaten van de opgaven op blz. 49. In dat laatste geval moet je wel nauwkeurig vermelden om welke opgave het gaat, en waarom het resultaat van die opgave van toepassing is.)

2. Tja, eigenlijk is het moeilijk schatten hoeveel programmeertalen er zijn, laat staan dat je kunt schatten hoeveel talen er ooit nog bij zullen komen. Laten we niet kinderachtig doen: laat $J_1, \ldots, J_N, \ldots$ een aftelbaar oneindige rij zijn van oude talen (Algol, PLI) bestaande talen (C, Java, C#, Ruby, ...) en talen die ooit nog zullen komen (C+++, C#+, ...). Is deze verzameling aftelbaar? Waarom (niet)?

De volgende stelling zal belangrijk worden als we het stop-probleem gaan bespreken (Hoofdstuk 17.7). We bespreken deze stelling nu al, omdat we er nu al het juiste instrumentarium voor hebben.

Stelling 5.2 (Gödel nummering) *Het is mogelijk elk programma $j \in J$ mechanisch te associëren met een uniek getal uit $\mathbb{N}$, het zg.* Gödel getal *van j.*

Bewijs:vrijv De nu volgende constructie is voor het eerst door o.a. de Oostenrijks-Amerikaanse logicus Kurt Gödel bedacht. (Er waren er meer die er in die tijd mee bezig waren. Het is niet helemaal duidelijk wie hier het eerst mee bezig was.)

Geef elk element uit A een uniek nummer. In ASCII is dat bijvoorbeeld: a: 97, b: 98, c: 99, ..., {: 173, |: 174, }: 175, ..., enzovoort. Het woord "hallo", bijvoorbeeld, is dan als volgt gecodeerd: 104-97-108-108-111. Dit is helaas niet voldoende om strings uniek te coderen, want 104-97-108-108-111 kan ook staan voor 10-49-71-08-108-1-11 (linefeed, 1, o, backspace, l, heading, vertical tab). Een unieke codering krijgen we wél als we dit rijtje getallen gebruiken als exponenten van de oneindige rij priemgetallen 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, ... Het woord "hallo" wordt dan bijvoorbeeld gecodeerd als

$$2^{104}3^{97}5^{108}7^{108}11^{111}$$

Dit is een belachelijk groot getal, maar wel uniek. Immers, ontbinding in priemfactoren is uniek (Stelling 1.7, blz. 13). Op deze manier kunnen alle strings gecodeerd worden.

De Gödel-nummering zorgt er voor dat verschillende strings worden gecodeerd als verschillende natuurlijke getallen. Bijvoorbeeld, het woord "halol" ($\neq$ "hallo") wordt gecodeerd als

$$2^{104}3^{97}5^{108}7^{111}11^{108} \neq 2^{104}3^{97}5^{108}7^{108}11^{111}$$

Omgekeerd correspondeert elk natuurlijk getal met precies één string uit A^+, ook weer omdat priemontbinding uniek is. □

Opgave 5.2 1. De laatste zin hierboven "Omgekeerd ... etc." moet worden genuanceerd. In principe is de bewering waar, maar dan moet er nog wel aan een voorwaarde zijn voldaan. Welke voorwaarde is dat?

2. Ga uit van ASCII. Met welke string zou

$$2^{104}3^{97}5^{9,761,451}7^{7,116,167}11^{9,761,666}$$

corresponderen? (We hebben dit getal alvast even voor je ontbonden.)

Als je het begrip Gödel-nummering snapt, dan snap je ook wel dat er in principe zeer veel manieren zijn om een Gödel-nummering aan te leggen. Maar de manier gebaseerd op de ASCII-nummering is wel de meest gebruikte.

In de volgende secties zullen we met de begrippen berekenbaarheid, beslisbaarheid en opsombaarheid aan de slag gaan.

5.3 Berekenbaarheid

Een functie heet berekenbaar als er een programma bestaat die die functie kan uitrekenen:

Definitie 5.3 (Berekening) *Zij $f : \mathbb{N} \to \mathbb{N}$ een partiële functie. We zeggen dat programma $j/1 \in J$ de functie f* berekent *als voor elke $n \in \mathbb{N}$ en elke uitvoering van $j(n)$ het volgende geldt:*

- *Als $f \downarrow n$, wordt $f(n)$ afgedrukt.*
- *Anders (als $f \uparrow n$) wordt **niets** afgedrukt.*

Opmerkingen:

1. Als j met een uitkomst $f(n)$ komt dan is j eigenlijk klaar, en mag het stoppen. Als j niet op deze manier werkt kunnen we j altijd veranderen in een programma j' dat altijd **stopt** na het afdrukken van een resultaat.
2. Als $f \downarrow n$ kan het op zich nog best lang duren voordat j de uitkomst $f(n)$ afdrukt. Er zit geen bovengrens op de rekentijd. Ook is de rekentijd van tevoren niet bekend.
3. Als j niets afdrukt, kan dat twee redenen hebben. Het kan zijn dat j heeft "ontdekt" dat $f \uparrow n$ en daarom stopt (of, "dommer", doorgaat zonder ooit nog wat af te drukken). Het kan ook zijn dat j gewoon nog bezig is en nog steeds niets heeft geconcludeerd.

[1] Eng.: computable, effectively computable.

Vind je het lastig of vervelend dat berekenbare functies in het algemeen partieel zijn? (We zijn immers gewend ons niet steeds af te vragen of een functie op een bepaald argument wel gedefinieerd is.) Als je het lastig vind, kunnen we dat begrijpen. Echter, partialiteit wordt van nature veroorzaakt door programma's die voor bepaalde invoer-waarden in een oneindige lus verzeild raken en dan geen (lees: een ongedefinieerd) resultaat opleveren.

Definitie 5.4 (Berekenbaar) *Een partiële functie $f : \mathbb{N} \to \mathbb{N}$ heet* berekenbaar*, of* effectief berekenbaar[1] *als er een programma $j/1 \in J$ bestaat welke f kan berekenen.*

Het adjectief "effectief" slaat op het algoritmische karakter van berekenbaarheid, dat wil zeggen, op het principe dat elke berekening typisch plaats vindt, of dient te vinden, in stapjes. Aan de andere kant maken veel (niet alle) informatici geen enkel onderscheid tussen de term "berekenbaarheid" enerzijds, en de term "*effectieve* berekenbaarheid" anderzijds.

Bijna alle functies $f : \mathbb{N} \to \mathbb{N}$ die we kennen zijn berekenbaar: optellen, aftrekken, kwadrateren, machtsverheffen, minimum nemen, maximum nemen, en ga zo maar door. Je kunt in gedachten vaak zelf wel voorstellen hoe je programma's voor elk van die functies kunt schrijven. En dat is dan meteen een bewijs voor hun berekenbaarheid!

Opgave 5.3 Is de functie met het volgende voorschrift berekenbaar? Waarom (niet)?

$$p : \mathbb{N} \to \mathbb{N} : n \mapsto \begin{cases} 1 & \text{als } n \text{ priem is,} \\ 0 & \text{anders.} \end{cases}$$

Het blijkt dus eenvoudig voorbeelden te geven van berekenbare functies in $\mathbb{N}^{\mathbb{N}}$. Moeilijker blijkt het een concrete *on*berekenbare functie aan te wijzen in $\mathbb{N}^{\mathbb{N}}$. Toch zijn deze er ook.

Stelling 5.3 (Onberekenbare functies) – *De verzameling $\mathbb{N}^{\mathbb{N}}$ bevat tenminste één onberekenbare functie.*

– *De verzameling $\mathbb{N}^{\mathbb{N}}$ bevat in feite* heel veel *onberekenbare functies.*

– *Meer in het bijzonder bevat $\mathbb{N}^{\mathbb{N}}$ meer onberekenbare dan berekenbare functies (!)*

Bewijs:vrijv (Het bewijs-verhaal is vrij lang, maar de twee achterliggende principes zijn in een oogwenk te begrijpen.)

Stel, er bestaat een zg. implementatie-functie

$$\theta : \mathbb{N}^{\mathbb{N}} \to J,$$

die elke functie, i.e., elk element van $\mathbb{N}^{\mathbb{N}}$, afbeeldt op een implementatie $j/1 \in J$. Dus $\theta(f)$ implementeert f, voor elke f. Het probleem is dat θ niet injectief kan zijn, immers J kan worden afgeteld, dus $|J| = |\mathbb{N}|$, terwijl $\mathbb{N}^{\mathbb{N}}$

over-aftelbaar is (Stelling 4.25, blz. 49). Dus $|\mathbb{N}^{\mathbb{N}}| > |J|$ wat inderdaad impliceert dat θ niet injectief kan zijn (het duiventil-principe, zie Opgave 4.17 op blz. 48). Per definitie betekent niet-injectiviteit dat er twee verschillende functies f en g zijn aan te wijzen, zó dat $\theta(f) = \theta(g)$, wat zou betekenen dat twee verschillende functies f en g door precies hetzelfde programma kunnen worden geïmplementeerd. Dat kan niet, dus onze aanname dat er een implementatie-functie θ bestaat blijkt onhoudbaar. Dit was het eerste deel van het bewijs.

Om te bewijzen dat $\mathbb{N}^{\mathbb{N}}$ meer onberekenbare dan berekenbare functies bevat, nemen we even aan dat de verzameling van onberekenbare functies even groot is als de verzameling van berekenbare functies. We nemen dus aan dat de verzameling van onberekenbare functies ook aftelbaar is. Van Opgave 4.19, blz. 48 weten we dat als twee disjunctie aftelbare verzamelingen worden verenigd, de vereniging ook weer aftelbaar is. En $\mathbb{N}^{\mathbb{N}}$ is niet aftelbaar maar over-aftelbaar. Dus onze aanname dat de verzameling van onberekenbare functies aftelbaar zou zijn, is fout. Dus de verzameling van onberekenbare functies is over-aftelbaar. □

Het bovenstaande bewijs is niet constructief: het zegt wel dat er onberekenbare functies zijn, maar het geeft geen concrete voorbeelden van onberekenbare functies. Misschien stelt je dit teleur: nu weet je nóg niet hoe zo'n onberekenbare functie er uit ziet, en kunnen we nog steeds geen vinger leggen op wat er dan zo moeilijk is aan onberekenbare functies dat je ze, per definitie van onberekenbaarheid, niet kunt implementeren.

We kunnen wel al een tipje van de sluier oplichten. De makkelijkst te identificeren onberekenbare functie is de zogenaamde *stop-functie*

$$H : \mathbb{N} \to \mathbb{N}.$$

(H van "halting".) De stop-functie bepaalt voor Gödel-nummeringen $G : J \to \mathbb{N}$ of programma $G(j/0)$ stopt, i.e:

$$H(G(j)) = 1 \Leftrightarrow \text{programma } j \text{ stopt (ooit).}$$

De stop-functie komt pas aan bod in Sectie 17.7 op blz. 217.

Een tweede alweer iets minder makkelijk te identificeren onberekenbare functie is de zogenaamde *busy beaver functie* $B : \mathbb{N} \to \mathbb{N}$. Deze bepaalt het maximaal aantal elementaire berekening-stappen dat mogelijk is bij een op den duur stoppend programma uit J ter lengte n:

$$B(n) = m \quad \Leftrightarrow \quad \text{er bestaat een } j \text{ met } |j| = n \text{ die tenminste } m \text{ berekeningstappen uitvoert en daarna stopt.}$$

Stel je bijvoorbeeld jezelf eens de vraag: "zou ik een programma in J kunnen schrijven, bestaande uit 100 karakters, dat bij uitvoering mijn PC langer dan 0.001 seconde bezighoudt? Langer dan 0.1? Langer dan 0.2?". (Om deze vraag direct te kunnen koppelen aan de busy beaver moet je eigenlijk nog weten hoeveel miljoen instructies één core uit je CPU per milliseconde aan kan.)

De busy beaver functie wordt hier verder niet besproken.

Opgave 5.4 Beargumenteer dat de Gödel-nummering zoals op blz. 60 uitgelegd, hoewel niet echt een functie van $\mathbb{N}$ naar $\mathbb{N}$, zelf ook berekenbaar is. Leg uit dat de berekenbaarheid beide kanten op werkt: van programma naar Gödel-nummer en andersom.

Er lijken er veel meer programma's in J dan berekenbare functies te zijn, een vermoeden dat door de volgende stelling wellicht nog eens wordt versterkt.

Stelling 5.4 (Padding lemma) *Elke berekenbare functie wordt geïmplementeerd door aftelbaar oneindig veel (syntactisch) verschillende programma's.*

Voor de goede orde: twee programma's zijn al syntactisch verschillend als ze één karakter verschillen.

Opgave 5.5 Bewijs Stelling 5.4. (Hint: let op de naam van de stelling, en denk aan spaties.)

Toch is een eventuele indruk dat er veel meer programma's dan berekenbare functies zijn, onterecht. Omdat J aftelbaar is (Stelling 5.1), is de verzameling berekenbare functies aftelbaar. Beide verzamelingen bezitten dus dezelfde kardinaliteit.

Wees ook altijd voorzichtig met de conclusie dat een functie berekenbaar is:

Opgave 5.6 Hieronder volgt een onware stelling, gevolgd door een fout bewijs. Kun je aangeven wat er fout gaat?

Stelling: iedere berekenbare functie $f : \mathbb{N} \to \mathbb{N}$ kan worden uitgebreid naar een totale functie, dat wil zeggen, naar een functie die op alle $n \in \mathbb{N}$ gedefinieerd is.

Bewijs: stel f wordt berekend door $j/1$. Schrijf nu het volgende programma j' dat voor iedere n de waarde $j'(n)$ als volgt berekent:

> Stel n is gegeven. Geef n aan j.
> – Als j stopt op n met resultaat $f(n)$, laat j' dan ook $f(n)$ teruggeven.
> – Als j niet stopt op n, laat j' dan 0 teruggeven.

De volgende stelling is wellicht wat wonderlijk, want tegenintuïtief. Immers, intuïtief gaat berekening maar één kant op. Denk bijvoorbeeld maar eens aan encryptie!

Stelling 5.5 *De inverse van een injectieve berekenbare functie bestaat, en is injectief en berekenbaar.*

Vergelijk deze bewering overigens met het resultaat van Opgave 3.35 op blz. 42.

Bewijs: Zij f een berekenbare functie, en laat $j \in J$ het programma zijn dat f uitrekent. Laat $n \in \mathbb{N}$ gegeven zijn waarvoor we f^{-1} willen uitrekenen.

We gaan de zogenaamde *zwaluwstaart-techniek* toepassen (Eng.: *dovetailing*). De zwaluwstaart-techniek is een manier om verschillende berekeningen quasi-parallel uit te voeren, door ze af te wisselen of op een andere manier door elkaar heen te vlechten. In dit geval voeren we $j(i)$ voor alle $i \in \mathbb{N}$ quasi-parallel uit op $\mathbb{N}$:

1. Eerst één stap van $j(0)$
2. Dan één stap van $j(0)$ en één stap van $j(1)$. De uitvoering van $j(0)$ is nu twee stappen ver.
3. Dan één stap van $j(0)$, één stap van $j(1)$, en één stap van $j(2)$. De uitvoering van $j(0)$ is nu drie stappen ver, de uitvoering van $j(1)$ is twee stappen ver, en de uitvoering van $j(2)$ is één stap ver.
4. Dan één stap van $j(0)$, één stap van $j(1)$, één stap van $j(2)$, en één stap van $j(3)$. De uitvoering van $j(0)$ is nu vier stappen ver, de uitvoering van $j(1)$ is drie stappen ver, de uitvoering van $j(2)$ is twee stappen ver, en de uitvoering van $j(3)$ is één stap ver.
5. ...

Als nu n verschijnt als functiewaarde van f, i.e., $n = f(m)$, voor zekere m, print dan m en stop. Het is belangrijk op te merken dat, als $f^{-1}(n)$ gedefinieerd is, deze ook uiteindelijk gevonden wordt, immers $f^{-1}(n)$ is gedefinieerd als en slechts als n in het bereik van f ligt. □

Opgave 5.7 1. Bespreek de gevolgen van deze stelling voor encryptie. Wat als encryptie injectief is? Surjectief?

2. Waarom zijn niet alle berekenbare functies inverteerbaar? Geef een (minimaal!) tegenvoorbeeld.

De items in de volgende opgave kunnen eenvoudiger bewezen worden.

Opgave 5.8 Bewijs:

1. De identiteitsfunctie is berekenbaar.
2. De compositie van twee berekenbare functies is berekenbaar.
3. Zij $f : \mathbb{N} \to \mathbb{N}$ (totaal en) bi-jectief en berekenbaar, en zij $g : \mathbb{N} \to \mathbb{N}$ onberekenbaar. Bewijs dat zowel $g \circ f$ als $f \circ g$ onberekenbaar zijn.

Opgave 5.9 Bekijk nog eens Opgaven 4.5 en 4.6 op blz. 4.5, met name de tekeningen die daarbij horen. Die tekeningen corresponderen niet onmiddellijk met een berekenbare aftelling.

i) Laat zien dat er een aftelling van $\mathbb{N} \times \mathbb{N}$ bestaat die berekenbaar is. (Hint 1: het is voldoende een injectief functievoorschrift $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ te geven. Grote kans dat het functievoorschrift dat je gegeven hebt, berekenbaar is. Hint 2: Gödel-nummering.)

ii) Geef commentaar op je gevonden functievoorschrift. Liggen de beelden van f ver uit elkaar? Is dat erg? Zo ja, wat zou je daar aan kunnen doen? Zo nee, waarom is dat niet erg?

iii) Met het antwoord van Opgave 5.5 (blz. 62) kunnen we concluderen dat f^{-1} berekenbaar is. Is het mogelijk een direct berekenbaar functievoorschrift van f^{-1} op te stellen?

5.4 Beslisbaarheid

We herhalen nog even de definitie van beslisbaarheid, zij het in iets andere vorm:

– Een programma $j/1 \in J$ kan *beslissen* over $X \subseteq \mathbb{N}$ als het voor elke $n \in \mathbb{N}$ kan uitmaken of $n \in X$.

– X wordt *beslisbaar*[2] genoemd als er een programma $j \in J$ bestaat dat over X kan beslissen.

We kunnen zonder beperking der algemeenheid aannemen dat j dan zó geschreven is dat het een 1 afdrukt als $n \in X$ en anders een 0. (Als j niet zo geschreven is, dan is het makkelijk voor te stellen hoe j kan worden omgeschreven naar een programma j' dat wel aan deze eisen voldoet.)

Opgave 5.10 Bewijs:

1. X is beslisbaar $\Leftrightarrow$ de karakteristieke functie

$$\chi(X) : N \to \{0, 1\}$$

   is berekenbaar.

2. Elke eindige verzameling $X \subseteq N$ is beslisbaar.
3. Als $X, Y \subseteq N$ beslisbaar zijn, dan zijn $X \cap Y$, $X \cup Y$, $X \setminus Y$, en X^c het ook.

Opgave 5.11 Laat zien dat

$$E = \{q \in \mathbb{Q} \mid q < e\}$$

beslisbaar is. (De constante e is Euler's constante.) Hint: vind twee oneindige rijen die e respectievelijk van boven en van onder benaderen.

Opgave 5.12 1. Beargumenteer dat het probleem of er aliens in het universum bestaan, beslisbaar is.

2. Beargumenteer dat de Goldbach-functie uit Voorbeeld 3.20, blz. 34, berekenbaar is.

Opgave 5.13 Bewijs dat een deelverzameling van de natuurlijke getallen, X, beslisbaar is als en alleen als X het bereik is van een totale (i.e., niet-partiële) niet-dalende berekenbare functie.

[2] Nederlands equivalent: recursief. Engelse equivalenten: decidable, recursive, recognisable by a Turing machine that always halts.

Laat ons nog een subtiel punt maken: de beslisbaarheid van een verzameling kan *non-constructief* worden bewezen, dat wil zeggen, zonder ooit een beschrijving van een beslissend algoritme te geven. Een klassiek voorbeeld is de verzameling

$$\Omega = \{n \in \mathbb{N} \mid \text{de decimale ontwikkeling van } \pi \text{ bevat tenminste } n \text{ opeenvolgende negens}\}$$

Deze verzameling is beslisbaar omdat tenminste één van de volgende alternatieven moet gelden (ga na!):

– $\Omega = \mathbb{N}$, en $\mathbb{N}$ is uiteraard beslisbaar.

– Ω is eindig, en eindige verzamelingen zijn beslisbaar. (Maak voor dat laatste zo nodig Opgave 2, blz. 63.)

Het vreemde is dat we de beslisbaarheid van Ω hebben aangetoond zonder ooit een programma te hebben gegeven dat voor zekere n bepaalt of ergens in de decimale ontwikkeling van π een string van n opeenvolgende negens voorkomt.

5.5 Opsombaarheid

We herhalen de definitie van opsombaarheid, zij het in een iets andere vorm:

> Een verzameling $X \subseteq \mathbb{N}$ heet *opsombaar*[3] als er een programma $j/0 \in J$ bestaat dat precies alle getallen in X afdrukt.

Voor een opsom-algoritme van X geldt het volgende.

– Het heeft geen input nodig.

– Het hoeft de elementen van X niet in een bepaalde volgorde af te drukken.

– Het mag elementen van X herhaald afdrukken.

– Er kunnen willekeurige lange en onvoorspelbare pauzes zitten tussen het afdrukken van getallen.

Opgave 5.14 Stel programma j somt een voor ons onbekende verzameling X op. In de eerste twee seconden worden 34 getallen afgedrukt. Toevalligerwijs (?) zijn dit de eerste 34 kwadraten. Het programma blijft lopen. Na tien minuten is er nog geen volgend getal afgedrukt. Na twee uur nog niets. Wat valt er over X te zeggen? Wat valt er over j te zeggen?

Elke opsombare verzameling is aftelbaar (waarom?) maar omgekeerd niet:

Stelling 5.6 *Er bestaan verzamelingen die aftelbaar maar niet opsombaar zijn.*

Bewijs:vrijv Later kunnen we dergelijke verzamelingen aanwijzen, maar op dit moment zullen we het met een non-constructief (ook wel: existentieel) bewijs moeten doen.

Alle deelverzamelingen van $\mathbb{N}$ zijn aftelbaar. Er zijn over-aftelbaar veel van dergelijke verzamelingen, immers, $|\mathbb{N}| < |2^{\mathbb{N}}|$). Volgens Stelling 5.1 zijn er "slechts" aftelbaar veel computerprogramma's. Die computerprogramma's kunnen dus ten hoogste aftelbaar oneindig veel deelverzamelingen van $\mathbb{N}$ opsommen. ("Ten hoogste", want veel computerprogramma's zijn helemaal niet met opsommen bezig. Er blijven overigens wel aftelbaar oneindig veel programma's over die wél met opsommen bezig zijn.) Er blijven in feite over-aftelbaar veel deelverzamelingen van $\mathbb{N}$ over die (dus wel aftelbaar maar) niet opsombaar zijn. □

Soms wordt opsombaarheid gedefinieerd als semi-beslisbaarheid ("half-bakken beslisbaarheid").

> X heet *semi-beslisbaar* als en slechts als er een programma $j/1 \in J$ bestaat dat stopt op precies alle elementen uit X.

(Dat wil zeggen, stopt op alle elementen van X, en niet stopt op alle elementen buiten X.)

Opgave 5.15 (Semi-beslisbaarheid) 1. Bewijs dat semi-beslisbaarheid inderdaad gelijkwaardig is aan de originele definitie van opsombaarheid. (Hint: gebruik de zwaluwstaart-techniek.)

2. Geef een definitie van opsombaarheid cq. semi-beslisbaarheid die zo veel mogelijk op de definitie van beslisbaarheid lijkt, en beargumenteer dat deze definitie inderdaad equivalent is aan opsombaarheid.

Er zijn veel equivalente definities van opsombaarheid. Eén daarvan gebruikt de notie van semi-karakteristieke functie. De *semi-karakteristieke functie* $\eta_X : \mathbb{N} \to \mathbb{N}$ van X is als volgt (gedeeltelijk!) gedefinieerd:

$$\eta_X(n) = \begin{cases} 0, & \text{als } n \in X, \\ \text{ongedefinieerd}, & \text{anders.} \end{cases}$$

De semi-karakteristieke functie is dus in het algemeen een partiële functie.

Hier volgen zoals gezegd een paar equivalente definities van opsombaarheid:

1. X is het domein van een berekenbare functie.

2. X is het bereik van een berekenbare functie.

3. De semi-karakteristieke functie van X is berekenbaar.

4. $X = \emptyset$ danwel het bereik van een totale berekenbare functie.

[3] Nederlands equivalent: enumereerbaar, semi-beslisbaar, semi-recursief. Engelse equivalenten: enumerable, recursively enumerable, r.e., computably enumerable, c.e., semi-recursive, semi-decidable, semi-computable, Turing-recognisable.

Bewijs:vrijv Laat we naar de oorspronkelijke definitie, Def. 5.2, verwijzen als definitie (0).

(0) $\Rightarrow$ (1), (3): veronderstel dat X wordt opgesomd door het programma j. Dan wordt de semi-karakteristieke functie η_X berekend door het volgende algoritme j':

> Laat n gegeven zijn. Voer j uit totdat n wordt afgedrukt. Als dat gebeurd, druk dan een 0 af en stop.

Merk op dat het domein van η_X nu inderdaad precies X is. Immers, het is X welke wordt opgesomd.

(1) $\Rightarrow$ (0): laat X het domein van een berekenbare functie f zijn. Neem aan dat f wordt uitgerekend door programma j. Dan kan X worden opgesomd door het volgende algoritme j':

> Laat n gegeven zijn. Gebruik de zwaluwstaart-techniek om j "quasi-parallel" op $\mathbb{N}$ uit te voeren:
>
> - eerst één stap van $j(0)$
> - dan één stap van $j(0)$ en één stap van $j(1)$ (de uitvoering van $j(0)$ is nu twee stappen ver)
> - dan één stap van $j(0)$, één stap van $j(1)$, en één stap van $j(2)$ (de uitvoering van $j(0)$ is nu drie stappen ver, de uitvoering van $j(1)$ is twee stappen ver, en de uitvoering van $j(2)$ is één stap ver)
> - ...
>
> Als nu alle n waarvoor $j(n)$ eindigt worden afgedrukt, dan krijgen we op deze manier een opsomming van X.

Omdat duidelijk (3) $\Rightarrow$ (1), is nu bewezen dat (0), (1) en (3) equivalent zijn.

Om (2) $\Rightarrow$ (1) te bewijzen is het voldoende om het zojuist besproken algoritme j' te veranderen in een algoritme j'' dat niet de argumenten n afdrukt bij stoppen van $j(n)$, maar in plaats daarvan de waarden $f(n)$ (die $j(n)$ immers berekent).

Blijft over te bewijzen (1) $\Rightarrow$ (2). We moet dus aantonen dat X het bereik is van een berekenbare functie. We weten inmiddels dat X het domein is van een berekenbare functie f. Als f wordt berekend door j, dan is X het bereik van een functie g die de waarde n aanneemt als $j(n)$ op den duur stopt, en anders ongedefinieerd is. Dus:

$$g(n) = \begin{cases} n & \text{als } j(n) \text{ op den duur stopt,} \\ \text{ongedefineerd} & \text{anders.} \end{cases}$$

Hier is een programma j''' welke functie g berekent:

> Laat n gegeven zijn. Voer $j(n)$ uit. Als $j(n)$ stopt, gooi het resultaat $f(n)$ weg, en druk in plaats daarvan de invoer n af.

We hebben nu bewezen dat (0), (1), (2) en (3) equivalent zijn. □

Opgave 5.16 Bewijs dat (4) equivalent is aan de andere definities van opsombaarheid.

Opgave 5.17 In veel opsombaarheidsbewijzen zul je een derde programma, j'', twee programma's j en j' gelijktijdig moeten laten uitvoeren, of beter: pseudo-gelijktijdig moeten laten uitvoeren, door hun uitvoering te vervlechten. Hier zijn twee methodes. Eén van deze twee methodes is goed en de andere is fout. Leg uit waarom.

1. Het programma j'' voert j en j' uit door *instructies* van j en j' door elkaar te vlechten: eerst één instructie van j, dan één instructie van j', dan één instructie van j, etc. Dat kan altijd, of j en j' nu verzamelingen opsommen of niet.
2. Als we weten dat zowel j als j' verzamelingen opsommen, dan kunnen we ze nog op een andere manier met elkaar vervlechten: we voeren j uit totdat deze een getal afdrukt. Daarna voeren we j' uit totdat deze een getal afdrukt. Daarna voeren we j uit totdat deze een getal afdrukt, etc.

Opgave 5.18 Zij X en Y opsombaar. Bewijs:

1. $X \cup Y$ is opsombaar.
2. (Ietsje moeilijker.) $X \cap Y$ is opsombaar. (Hint: het te construeren programma zal meer moeten onthouden.)

Soms worden zg. *non-deterministische algoritmen* beschouwd. Dit zijn algoritmen, of programma's, waar het is toegestaan om op sommige plekken variabelen een willekeurige waarde te geven:

$$x := \text{rand}(\mathbb{N});$$

Opgave 5.19 Bewijs: een verzameling $X \subseteq \mathbb{N}$ is opsombaar als en slechts als de elementen van X kunnen verschijnen in de uitvoer van een non-deterministisch nul-plaatsig programma $j/0$.

Opgave 5.20 Zij X en Y opsombaar. Bewijs: $X \times Y$ is opsombaar.

Opgave 5.21 Zij $X_1, \ldots, X_n$ opsombaar. Bewijs:

1. $\cup_{i=1}^{n} X_i = X_1 \cup \cdots \cup X_n$ is opsombaar. (Hint: gebruik het resultaat van Opgave 5.18.)
2. $\cap_{i=1}^{n} X_i = X_1 \cap \cdots \cap X_n$ is opsombaar.
3. $\prod_{i=1}^{n} X_i = X_1 \times \cdots \times X_n$ is opsombaar.
4. Alle (eindige) verzamelingstheoretische combinaties van $X_1, \ldots, X_n$ zijn opsombaar, zolang er gecombineerd wordt met operatoren uit $\{\cup, \cap, \times\}$.

Opgave 5.22 Zij $X_1, \ldots, X_n, \ldots$ opsombaar. Bewijs:

1. $\cup_{i=1}^{\infty} X_i$ is opsombaar. (Hint: gebruik de zwaluwstaart-techniek.)

2. $\prod_{i=1}^{\infty} X_i$ is niet opsombaar, zelfs niet als alle X_i eindig zijn.

Opgave 5.23 Zij $X_1, \ldots, X_n, \ldots$ opsombaar. Is $\cap_{i=1}^{\infty} X_i$ opsombaar?

5.6 De stelling van Post

Er is uiteraard een relatie tussen opsombare en beslisbare verzamelingen. De stelling van Post gaat precies daarover.

De Pools-Amerikaans wiskundige Emil Post (1897-1954) was, naast Turing, één van de meest belangrijke grondleggers van de informatica. Post werd geboren in Polen en emigreerde als kind naar Amerika. Z'n (werk-) colleges verliepen, op z'n zachtst gezegd, ongebruikelijk:

> *Post's classes were tautly organised affairs. Each period would begin with student recitations covering problems and proofs of theorems from the day's assignment. These were handed out apparently at random and had to be put on the blackboard without the aid of textbooks or notes. Woe betide the hapless student who was unprepared. He (or rarely she) would have to face Post's "more in sorrow than anger look". In turn, the students would recite on their work. Afterwards, Post would get out his 3-by-5 cards*[4] *and explain various fine points. The class would be a success if he completed his last card just as the bell rang. Questions from the class were discouraged: there was no time. Surprisingly, these inelastic pedagogic methods were extremely successful, and Post was a very popular teacher.* (M. Davis, *Emil L. Post: his life and work*, herdruk 1994.)

Voordat we de stelling van Post bewijzen eerst een definitie.

Definitie 5.5 (Complementair opsombaar) *Een verzameling $X \subseteq \mathbb{N}$ heet* complementair opsombaar *als het complement opsombaar is.*

Alternatieve termen: co-opsombaar, co-enumereerbaar, co-recursief enumereerbaar, co-recursively enumerable, co-r.e.

In zekere zin zijn complementair opsombare verzamelingen nog pathologischer, of ongrijpbaarder, dan opsombare verzamelingen. Bij een opsombare verzameling X kon je in ieder geval nog wachten totdat je favoriete getal x in de opsomming verscheen om te mogen concluderen dat $x \in X$. Bij een complementair opsombare verzameling Y is dat niet mogelijk. Als j een algoritme is dat het complement van Y opsomt en een getal x afdrukt, dan kunnen we alleen maar concluderen dat $x \notin Y$. Het doet denken aan het tekenen van een plaatje door stipjes te zetten op de achtergrond. Omdat we geen stipjes mogen zetten op de plek van het plaatje zelf, kunnen we alleen maar conclusies trekken over de vorm van het plaatje door naar de stipjes in de achtergrond te kijken. Met een beetje verbeeldingskracht kunnen complementair opsombare verzamelingen wel worden gezien als de "zwarte gaten" van de berekenbaarheidstheorie.

[4] Gelinieerde kartonnen index-kaarten, met een grootte van 3x5 inch.

Stelling 5.7 (Stelling van Post) *$X \subseteq \mathbb{N}$ is beslisbaar $\Leftrightarrow$ X is zowel opsombaar als complementair opsombaar.*

Dit is een belangrijke stelling. Veel belangrijke stellingen zijn moeilijk te formuleren en moeilijk te bewijzen. Andere belangrijke stellingen zijn makkelijk te formuleren maar moeilijk te bewijzen (denk aan de grote stelling van Fermat). Deze belangrijke stelling is makkelijk te formuleren én makkelijk te bewijzen. Je mag dus zelf proberen het bewijs te geven.

Opgave 5.24 Bewijs de stelling van Post.

Opsombare verzamelingen kunnen helemaal gedefinieerd worden in termen van beslisbaarheid:

Stelling 5.8 *Een verzameling $X \subseteq \mathbb{N}$ is opsombaar als en slechts als het de projectie is van een beslisbare deelverzameling van $\mathbb{N} \times \mathbb{N}$.*

Bewijs: De projectie van een opsombare verzameling is opsombaar (verwijder telkens het eerste element), dus de projectie van een beslisbare verzameling is dan zeker opsombaar.

Stel, omgekeerd, dat verzameling X wordt opgesomd door programma j. X is de projectie van een beslisbare verzameling, Y, van paren, (x,n), waarbij het eerste element, x, verschijnt tijdens de eerste n stappen van j. (De eigenschap van een paar (x,n) is evident beslisbaar.) □

Het tweede gedeelte van het bewijs is behoorlijk subtiel, en geeft precies de connectie aan tussen opsombaarheid en beslisbaarheid. Er is ook een sterke relatie tussen opsombaarheid en berekenbaarheid:

5.7 Opsombaarheid en berekenbaarheid

We hebben gezien dat de notie van een opsombare verzameling kan worden gedefinieerd in termen van berekenbare functies. Deze situatie kan worden omgedraaid.

Stelling 5.9 *Een functie $f : \mathbb{N} \to \mathbb{N}$ is berekenbaar als en slechts als haar grafiek*

$$F = \{(x,y) \mid x,y \in \mathbb{N}, f \downarrow x, f(x) = y\}$$

opsombaar is.

Bewijs: Veronderstel dat f berekenbaar is. Dan bestaat er een programma $j/0$ dat het domein van definitie van f, i.e., de verzameling

$$\text{domdef}(f) = \{n \in \mathbb{N} \mid f \downarrow n\}$$

opsomt. Schrijf nu het volgende programma $j'/0$:

> Laat j de verzameling domdef(f) opsommen.
> Druk in plaats van de n het paar $(n, f(n))$ af.

Als er, omgekeerd, een programma j bestaat dat F opsomt, dan kan f als volgt worden berekent.

> Gegeven is n. Voer j uit en wacht op een
> eventueel verschijnen van het paar (n, y).
> Druk, zodra dat gebeurt, de tweede coördinaat
> y af, en stop.

Zij $f : \mathbb{N} \to \mathbb{N}$ een partiële functie, en $X \subseteq \mathbb{N}$. Het *beeld* van X onder f, is gedefinieerd als

$$f[A] = \{f(n) \mid n \in X \text{ en } f \downarrow n\}.$$

Het *inverse beeld* van X onder f (Eng.: *pre-image*), is gedefinieerd als

$$f^{-1}[X] = \{n \mid f \downarrow n \text{ en } f(n) \in X\}.$$

Stelling 5.10 *Het beeld en inverse beeld van een opsombare verzameling onder een berekenbare functie zijn beiden ook weer opsombaar.*

Bewijs:vrijv Om een invers beeld van een opsombare verzameling X onder een berekenbare functie te krijgen, volstaat het de grafiek van f te doorsnijden met de opsombare verzameling $\mathbb{N} \times X$ om dan vervolgens de projectie van de deze doorsnijding op de eerste coördinaat te nemen. Een analoog argument met de coördinaten verwisseld is van toepassing op het beeld van f. □

Opgave 5.25 Zij X een beslisbare verzameling. Toon aan dat $Y = \{2x \mid x \in X\}$ ook beslisbaar is. (Waarschuwing: let goed op in welke richting je converteert!)

Opgave 5.26 Zij X en Y twee opsombare verzamelingen met niet-lege doorsnede. Bewijs dat er opsombare disjuncte verzamelingen X' en Y' zijn, zo dat

$$X' \subseteq X,\ Y' \subseteq Y, \text{ en } X' \cup Y' = X \cup Y.$$

Opgave 5.27 Bewijs dat iedere oneindige opsombare verzameling kan worden gerepresenteerd als het beeld van een totale berekenbare injectieve functie. (Hint: verwijder doublures in de oorspronkelijke opsomming.)

Opgave 5.28 Bewijs dat iedere oneindige opsombare verzameling een oneindige beslisbare verzameling bevat. (Hint: gebruik de oplossing van het voorgaande probleem en kies een stijgende deelrij.)

Opgave 5.29 Bekijk evt. Opgave 4.31 op blz. 50. Als f een Diofantische vergelijking is met k variabelen, $k \geq 1$ dan heet

$$X = \{x_1 \mid \exists x_2, \ldots, x_k : f(x_1, x_2 \ldots, x_k) = 0\}$$

per definitie een *Diofantische verzameling*. Een Diofantische verzameling is dus een projectie van de nulpunten van een meervariabelige veelterm in één coördinaat. Voorbeelden:

– Twee-machten: $\{x_1 \mid \exists x_2 : x_1 - x_2^2 = 0\}$.

– $\{x_1 \mid x_1 - (2x_2 + 3)x_3 = 0\}$.

Opmerkingen:

- $X \subseteq \mathbb{N}$.
- Het is mogelijk dat X leeg is of juist gelijk is aan heel $\mathbb{N}$.
- Omdat x_1 zonder problemen kan worden verwisseld met elke andere variabele, doet het er eigenlijk niet toe dat in de bovenstaande definitie de specifiek eerste variable geïsoleerd wordt. Het isoleren van een andere variabele geeft geen nieuwe of andere Diofantische verzamelingen.

Tussen 1950 en 1970 bewezen Yuri Matiyasevich, Julia Robinson, Martin Davis en Hilary Putnam (kortweg: MRDP) iets heel speciaals, namelijk dat elke opsombare verzameling Diofantisch is! Zij bewezen dus dat bij iedere opsombare verzameling een meervariabelige veelterm moet bestaan waarvan de eerste coördinaten van alle nulpunten precies gelijk is aan die opsombare verzameling!

Hoe spectaculair dit is blijkt uit een aantal willekeurige voorbeelden. Noem maar een willekeurige opsombare verzameling. De verzameling priemgetallen, bijvoorbeeld. Deze is evident opsombaar. (Ga na!) Uit de MRDP-stelling volgt nu onmiddellijk dat er een veelterm moet bestaan waarvan de eerste coördinaten van alle nulpunten precies de verzameling van alle priemgetallen vormt! (Zo'n veelterm is trouwens daadwerkelijk geconstrueerd, en makkelijk te vinden op internet.)

Nog belangrijker en het uiteindelijke doel van de MRDP-stelling is dat hiermee Hilbert's 10e probleem is beantwoord, en wel in negatieve zin.

1. Bewijs zelf het omgekeerde, namelijk dat elke Diofantische verzameling opsombaar is.
2. Uit het vorige antwoord en de MRDP-stelling volgt dat er geen algoritme kan bestaan voor het vinden van nulpunten voor willekeurige Diofantische vergelijkingen. Beargumenteer waarom dat zo is.

Opgave 5.30 (Grote stelling van Fermat) Omstreeks 1995 bewees de Engelse wiskundige Andrew Wiles (geboren 1953) samen met andere wiskundigen dat de vergelijking

$$x^n + y^n = z^n$$

alleen voor $n \in \{1,2\}$ geheeltallig oplossingen heeft. Denk aan voorbeeldjes van de stelling van Pythagoras op de middelbare school, waarbij $n = 2$ en typisch $x = 3$, $y = 4$ en $z = 5$. (We schrijven "omstreeks 1995" omdat na een eerste publicatie het bewijs nog lacunes bevatte en er vervolgens nog ongeveer een jaar aan is gewerkt om het te repareren.)

Zoals je misschien weet heeft de grote stelling van Fermat, soms ook wel de laatste stelling van Fermat genoemd, sinds zijn formulering in 1637 talloze professionele en amateur-wiskundigen tot de rand van de waanzin gebracht. De stelling is zeer makkelijk te formuleren, maar het bewijzen ervan bleek andere koek. Het probleem werd nog uitdagender omdat Fermat tijdens het lezen van Brachet's vertaling van Diophantos' "Arithmetica" zelf aantekende een heel opmerkelijk bewijs te hebben gevonden, waarvoor, volgens Fermat, de kantlijn echter net te weinig ruimte bood. Veel wiskundigen betwijfelen of Fermat ook inderdaad een bewijs gevonden had. Wat denk jij?

1. Bewijs dat de verzameling

$$\text{Fermat} = \{n \in \mathbb{N} \mid x^n + y^n = z^n \text{ bezit geheeltallig oplossingen}\}$$

   opsombaar is. (Hint: vanwege Wiles resultaat is dit onderdeel eenvoudig te beantwoorden.)

2. Dezelfde vraag, maar nu met losse handen (i.e., zonder gebruikmaking van de grote stelling van Fermat) >:-).

Voor de volgende opgave is het begrip "pseudo-inverse" nodig.

Definitie 5.6 (Pseudo-inverse) *Zij $f : \mathbb{N} \to \mathbb{N}$ een partiële functie. Een* pseudo-inverse *van f is een partiële functie $g : \mathbb{N} \to \mathbb{N}$, zó dat* $\text{dom}(g) = \text{ran}(f)$ *en $f(g(f(x))) = f(x)$ voor alle $x \in \mathbb{N}$ met $f \downarrow x$.*

Het begrip pseudo-inverse is gewoon een generalisatie van het begrip "inverse functie", zodanig dat dit begrip ook nog betekenis heeft voor partiële functies.

Opgave 5.31 Toon aan dat elke berekenbare functie f een pseudo-inverse heeft.

5.8 Hyper-berekenbaarheid

In de inleiding werd de Church-Turing these besproken. Deze poneert dat berekenbaarheid wordt vertegenwoordigd door een grote klasse van even krachtige mechanismen zoals μ-recursieve functies, Turing machines, λ-calculus, Markov-algoritmen, Post-kanonieke systemen, Conway's game-of-life implementaties, register-machines, en nagenoeg alle hedendaagse programmeertalen.

We legden ook uit dat de Church-Turing these, behalve een hypothese, ook een *uitdaging* is om met een mechanische beslissingsprocedure te komen die niet μ-recursief is (of niet kan worden berekend met een Turing-machine, of niet kan worden berekend met een register-machine, etc.) Men kan wel stellen dat deze uitdaging een groot aantal informatici heeft geïnspireerd.

Onderzoek naar niet-recursieve beslissingsprocedures is op dit moment in volle gang. Dit onderzoek wordt verricht onder ronkende namen als *hyper-berekenbaarheid*, *super-recursiviteit* en *super-Turing berekenbaarheid*. Onderzoek hiernaar is moeilijk, omdat buiten de conventionele klasse van equivalente berekeningsmodellen de ideeën over hyper-berekenbaarheid snel divergeren. Immers, het bindende raamwerk van de Church-Turing hypothese ontbreekt. Ook zijn er andere valkuilen, zoals spraakverwarring en dilettantisme (spreken of schrijven over hyper-berekenbaarheid, zonder je eerst goed te hebben verdiept in de notie van berekenbaarheid zelf).

Grofweg valt hyper-berekenbaarheid uiteen in twee categorieën: noties van hyper-berekenbaarheid die fysiek realiseerbaar maar praktisch onbruikbaar zijn, en noties van hyper-berekenbaarheid die in principe bruikbaar, maar vooralsnog fysiek onrealiseerbaar lijken te zijn.

1. Voorbeelden uit de eerste categorie zijn berekening door non-determinisme, asynchroniciteit en software-updates. (Bij het laatste voorbeeld wordt hyper-berekenbaarheid veroorzaakt de interventie van menselijke creativiteit.)

   Met enige kennis van maat- en integratietheorie kan bijvoorbeeld makkelijk worden bewezen dat, als je twee ongesynchroniseerde programma's die beiden nullen en enen afdrukken, samen deze nullen en enen op een

tape laat schrijven, er met kans 1 een onberekenbaar getal wordt afgedrukt. Echter, we hebben niets aan dit resultaat want er kan van tevoren niet worden bepaald welk onberekenbaar getal dat is.

2. Voorbeelden uit de laatste categorie zijn berekeningen met echte reële getallen (i.p.v. floating point benaderingen), Blum-Shub-Smale machines, en quantum-computers.

Bezint eer ge begint. Lees ook de sectie over cranks op blz. 16.

Deel II

Propositielogica

Hoofdstuk 6

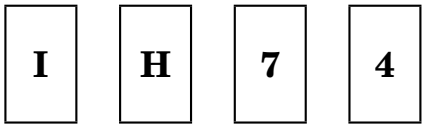
Logica

6.1 Wat is het en heb je eraan?

De gangbare omschrijving van 'logica' is: leer van het correct redeneren. Theorievorming over de manier waarop mensen *zouden moeten* redeneren heeft al vanaf de Klassieke Oudheid plaatsgehad. Er is veel voor te zeggen om de logica te beschouwen als het verst ontwikkelde onderdeel van de filosofie. Logica is een onderdeel van de filosofie (of: 'wijsbegeerte') omdat het een vak is waar mensen die zich 'filosoof' noemden zich al vanaf de Oudheid hebben beziggehouden. Er heeft sinds de syllogismen-leer van Aristoteles (384–322 voor Christus) een duidelijke vooruitgang plaatsgehad in dit vak. Die ontwikkeling is weliswaar met horten en stoten verlopen, maar zij is spectaculairder dan die in welk ander onderdeel van de filosofie ook.

Logica onderzoekt niet het feitelijk redeneren van mensen. Het is geen *empirische* wetenschap die bij voorbeeld probeert uit te zoeken waarom mensen vaak een bepaald soort (feitelijk constateerbare) vergissingen maken. Het vak logica is *normatief*: het onderzoekt de normen voor het correct redeneren. We geven een voorbeeld om het verschil aan te geven tussen een empirische en een normatieve aanpak van een redeneerprobleem.

Stel je voor dat er vier kaarten op tafel liggen, en je weet dat elke kaart aan de ene kant een letter vertoont, en aan de andere kant een cijfer. De kaarten liggen als volgt:



De vraag is nu: welke twee kaarten moet ik omdraaien om de volgende bewering op juistheid te controleren:

> ***Bewering.*** *Van de kaarten die op tafel liggen hebben die met een* klinker *op de voorzijde een* even getal *op de achterzijde?*

Uit tests is gebleken dat veel mensen (*ook* mensen die bij voorbeeld taalkunde, informatica of wiskunde hebben gestudeerd) bij het beantwoorden van deze vraag een bepaalde elementaire redeneerfout maken. Wie dit soort tests doet houdt zich bezig met een vorm van empirische wetenschap (bij voorbeeld: experimentele psychologie), niet met logica. De logicus is alleen geïnteresseerd in het blootleggen van het patroon dat achter de *correcte* redenering ligt.

Tot ver na de Middeleeuwen is gedacht dat het vak logica in handen van Aristoteles zijn definitieve vorm had gekregen. Die opvatting wordt zelfs nog verkondigd in Kant's *Kritik der reinen Vernunft*: zie het voorwoord in [Kant 1787]. Het zal dan nog bijna een eeuw duren voor het vak weer een grote sprong voorwaarts maakt. In 1879 verschijnt Gottlob Frege's *Begriffsschrift*, waarin in 88 bladzijden een symbooltaal wordt geïntroduceerd waarmee je alle redeneringen die Aristoteles had behandeld kunt analyseren, plus nog veel meer (zie [Frege 1879]). De door Frege geïntroduceerde taal is in feite die van de predikatenlogica, waarmee we in hoofdstuk 6 zullen kennismaken.

Frege's impuls heeft ervoor gezorgd dat logica niet alleen voor (taal)filosofen maar ook voor wiskundigen interessant werd. Het logische onderzoek van wiskundige redeneersystemen is intussen uitgegroeid tot een aparte tak van wiskunde, de zogenaamde meta-mathematica. Zie voor meer informatie: [Van Dalen 1978].

Wij blijven voorlopig wat dichter bij huis, door de vraag te stellen hoe de symbooltalen die door logici zijn voorgesteld zich verhouden tot de natuurlijke taal. In vergelijking met natuurlijke taal hebben formele talen het voordeel van een absolute precisie. Ze hebben echter ook een groot nadeel: hun gebrek aan flexibiliteit. Frege vergeleek in zijn *Begriffsschrift* de verhouding tussen natuurlijke taal en formele taal met die tussen het blote oog en de microscoop, elk met hun eigen toepassingsgebied.

De formele talen die door logici zijn ontwikkeld verschillen onderling sterk in uitdrukkingskracht. De uitdrukkingskracht van een logische taal hangt af van het scala van logische *operatoren* dat in die taal voorkomt. De eenvoudigste (en dus minst krachtige) logische taal is die van de propositielogica. Deze taal kent alleen *logische voegwoorden* of *connectieven*, waarmee zinnen onderling verbonden kunnen zijn, zoals *en* en *of*. De eenvoudigste elementen die worden onderscheiden zijn de *basiszinnen*. De interne structuur van zo'n basiszin wordt niet nader beschouwd; daar is het analyse-apparaat niet op berekend. Een zin als "Iedereen slaapt"moet dus worden beschouwd als een ongeanalyseerd geheel. Zodra we ook zogenaamde *kwantoren* toevoegen aan ons logische arsenaal—hetgeen gebeurt in de predikatenlogica—kunnen we de structuur van deze zin verder blootleggen. De vertaling in de predikatenlogica is: $\forall x S x$.

De predikatenlogica kan nog weer verder worden uitgebreid, zodat een nog krachtiger analysemiddel ontstaat. In de *modale* predikatenlogica kun je ook de logische functie van *modale hulpwerkwoorden* en *modale adverbia* uitdrukken, doordat zogenaamde modale operatoren $\Diamond$ ("het is mogelijk dat") en $\Box$ ("het is noodzakelijk dat") aan de taal zijn toegevoegd. De

vertaling van “Niemand kan slapen” kan nu worden: $\forall x \neg \Diamond Sx$.

De meest krachtige formele taal is echter niet zonder meer de beste. Formalisering is geen doel op zichzelf, maar een *middel* om een bepaald doel te bereiken: de verheldering van de rol die de structuur van een bewering heeft in verband met de geldigheid of ongeldigheid van redeneringen waarin die bewering voorkomt. Waar het gaat om simpele redeneringen heeft het opstellen van zwaar geschut geen zin. Op de manier waarop in de logica redeneringen worden geanalyseerd gaan we in § 6.2 nader in.

Tot besluit van deze inleidende paragraaf wat relevante literatuur. Wie het vak logica geplaatst wil zien tegen de achtergrond van andere onderdelen van de filosofie, leze [Van Eijck 1982]. De rol van de logica in het alledaagse denken wordt belicht in [Emmet 1969]. Er bestaan vele inleidingen in de logica. Ze variëren qua oriëntatie van algemeen-filosofisch tot mathematisch. Het meest toegespitst op verbanden met de taalkunde is [Gamut 1982, twee delen], van harte aanbevolen. Wiskundig getinte inleidingen in de logica kun je vinden in [Van Dalen 1983] en [Enderton 1972].

6.2 Redeneringen en redeneerschema’s

Een redenering—mits voldoende gestroomlijnd—is op te vatten als een rijtje van *uitgangspunten* of *premissen*, gevolgd door een *conclusie*. Als we de conclusie scheiden van de premissen door een streep, dan kunnen we een redenering zo weergeven:

Als Jan boos is komt hij niet.
Jan is gekomen.

Jan is niet boos.

Boven de streep staan de premissen, eronder staat de conclusie. Soms is er maar één premisse, bij voorbeeld in de redenering die je nodig hebt om het klinker-getallen-probleem uit § 6.1 op te lossen:

Voor elke kaart geldt:
als een klinker, dan ook een even getal.

Voor elke kaart geldt:
als een oneven getal, dan ook een medeklinker.

We zullen hier niet ingaan op de functies van redeneringen in het dagelijks leven (waar de redeneringen vaak achteraf worden toegevoegd aan wensen of standpunten die al vast stonden, bij wijze van ideologische rechtvaardiging). Het zij slechts opgemerkt dat de redeneringen in systematische denkdisciplines—zoals daar zijn de filosofie, de theologie, de wiskunde, de informatica en de taalwetenschap—een dankbaarder object van studie vormen dan die uit de wereld van alledag. Een conclusie over het nut van logica bij het vinden van je weg in dit leven moet je zelf maar trekken...

In de logica wordt bij het bestuderen van redeneringen geabstraheerd van de inhoud. De volgende definitie speelt in dat abstractieproces de hoofdrol:

Definitie 6.1 *Een redenering heet* **geldig** *wanneer de waarheid van de premissen de waarheid van de conclusie afdwingt.*

Let op: de definitie zegt niets over de feitelijke waarheid van de premissen en de conclusie. Voor de geldigheid van een redenering is voldoende dat aan de volgende eis voldaan is: *als* de premissen waar zijn, *dan ook* de conclusie. De premissen kunnen dus best onwaar zijn, terwijl de redenering geldig blijft. Aan de andere kant kunnen we ook de situatie hebben dat de premissen en de conclusie allebei waar zijn, terwijl we niettemin met een ongeldige redenering van doen hebben. De premissen en de conclusie zijn dan alleen ‘toevallig’ waar, om zo te zeggen. Hier is een voorbeeld van een ongeldige redenering met ware premissen en een ware conclusie:

Alle muizen hebben staarten.
Alle muizen zijn knaagdieren.

Alle knaagdieren hebben staarten.

De redenering is niet geldig, want andere redeneringen volgens ditzelfde stramien kunnen de combinatie vertonen van ware premissen met een onware conclusie. In de volgende variant op bovenstaande redenering hebben we de *inhoudswoorden* vervangen en de *functiewoorden* gehandhaafd. Deze procedure levert een redenering op waarvan de premissen waar zijn maar de conclusie niet.

Alle feministes zijn vrouwen.
Alle feministes zijn mensen.

Alle mensen zijn vrouwen.

Bij geldige redeneringen zal zoiets je niet overkomen, mits je bij het vervangen van onderdelen in de premissen en de conclusie het stramien van de redenering maar intact laat.

Blijkbaar doet het er voor de geldigheid van een redenering helemaal niet toe waar de redenering over gaat, maar alleen welk stramien hij heeft. We stappen daarom over van concrete redeneringen naar *redeneerschema’s*, door stelselmatig te abstraheren van de elementen die er niet toe doen. Zo zien we dat de eerste voorbeeldredenering van deze paragraaf verloopt volgens het schema:

Als A, dan niet B.
B.

Niet A.

De letters A en B staan voor willekeurige beweringen (of: proposities). Door voor A “Jan is boos” en voor B “Jan komt” in te vullen krijgen we de bovenstaande redenering weer terug. Door voor A “Marie is verontwaardigd” en voor B “Marie groet Piet” in te vullen krijgen we een andere geldige redenering, enzovoorts.

Het schema van de redenering uit het kaarten-voorbeeld is:

Alle K zijn E.

Alle niet-E zijn niet-K.

Nu staan K en E voor uitdrukkingen die eigenschappen uitdrukken (in het jargon dat we in § 1.4 hebben ingevoerd: “die een eigenschap als denotatie hebben”). We kunnen ook andere eigenschapswoorden invullen, en we krijgen dan een andere geldige redenering (vul bij voorbeeld “kinderen” en “engelen” in).

Het schema van het muizen-voorbeeld is:

Alle P zijn Q.
Alle P zijn R.

Alle Q zijn R.

Merk op dat we tussen neus en lippen door een beetje hebben gestroomlijnd: R kan worden vervangen door “staartdier”, en we krijgen “zijn staartdieren” in plaats van “hebben staarten”. Je ziet dat het analyseren soms enige souplesse vraagt: de analyses zijn steeds *ad hoc*, en er wordt geen *systematisch* verband gelegd tussen natuurlijke-taal zinnen en logische vormen. In systemen die wel zo’n systematisch verband pretenderen te geven, zoals de Logische Vorm theorie binnen de Transformationele taalkunde of de Montague grammatica zijn zulke losse-pols-manoeuvres verboden.

Een stel beweringen die een bepaald schema vertonen, maar zo dat de premissen waar zijn en de conclusie niet, heet een *tegenvoorbeeld* tegen het schema. Het voorbeeld met de feministes hierboven, is een tegenvoorbeeld tegen het laatstgenoemde schema. We kunnen nu ook zeggen: een redeneerschema is geldig desda er geen tegenvoorbeelden tegen bestaan.

Enkele redeneerschema’s zijn zo beroemd dat ze speciale namen hebben gekregen. De volgende schema’s heten respectievelijk *Modus Ponens* en *Modus Tollens*.

Als A dan B.
A.

B.

Als A dan B.
Niet B.

Niet A.

Je kunt proberen wat je wilt om tegenvoorbeelden tegen deze redeneerschema’s te geven, het zal je niet lukken: de schema’s zijn geldig.

Opgave 6.1 Geef een tegenvoorbeeld tegen het volgende schema:

Als A dan niet B.

Als niet A dan B.

Opgave 6.2 Geef een tegenvoorbeeld tegen het volgende schema:

Als niet A dan B.
B.

Niet A.

6.3 Logica en games

Enkele woorden over de relatie tussen logica en (computer) games. Als je gametechnology studeert is het misschien moeilijk je te motiveren voor een vak als logica, omdat er geen verband tussen deze twee onderwerpen lijkt te bestaan. Is dat zo? Is er geen verband? Het antwoord ligt genuanceerd.

Laten we beginnen te vermelden dat er academisch onderzoek wordt gedaan naar de toepassing van logica in spelen. Echter, dit betreft voornamelijk onderzoek naar logica in klassieke speltheorie, zoals bijvoorbeeld schaken, backgammon, poker, gecompliceerde versies van het zogenaamde prisoners dilemma, en meer in het algemeen economische speltheorie zoals dat in 1950 en later is bedacht door Von Neumann en anderen (lees bijvoorbeeld het artikel in de online Stanford Encyclopedia of Philosophy “Logic and Games”, of stel je op de hoogte van Van Benthem’s onderzoek naar logic and games). Of deze spelen veel met real-time computer games te maken hebben valt inderdaad te betwijfelen. Uiteraard is dat in dat onderzoek ook niet de bedoeling.

De vraag blijft of er dan zoiets is als logica voor real-time games. Tot op zekere hoogte is dat er wel, maar veel minder, en als het er is, dan gaat het veel meer in de richting van klassieke speltheorie, en maakt het gebruik van continue i.p.v. discrete wiskunde. Met name maakt real-time game theory gebruik van differentiaalvergelijkingen. De theorie van differentiaalspelen (Eng.: *differential games*), bijvoorbeeld, houdt zich bezig met strategisch gedrag van spelers in $\mathbb{R}^2$, $\mathbb{R}^3$, of generalisaties daarvan. (Denk bijvoorbeeld aan vliegbewegingen van straaljagers in een dogfight.) De bestudering van deze theorie werd vooral in de jaren ’50 en ’60 van de vorige eeuw heftig gesubsidieerd in het kader van de koude oorlog tussen de toenmalige Sovjet-Unie en de VS.

Hoofdstuk 7

Propositielogica

7.1 Voegwoorden en hun betekenis

"Jan is boos en hij is verdrietig" bestaat uit twee zinnen, te weten de zin "Jan is boos" en de zin "hij is verdrietig". Deze twee zinnen zijn door middel van het voegwoord *en* verbonden tot een samengestelde zin. Wanneer we even wat stroomlijnen door "hij" te vervangen door "Jan" krijgen we twee zinnen, "Jan is boos" en "Jan is verdrietig", waarvoor geldt dat informatie over het waar of onwaar zijn ervan (logisch jargon: informatie over hun *waarheidswaarde*) voldoende is om vast te stellen of de samengestelde zin al dan niet waar is. Anders gezegd: de waarheidswaarde van "Jan is boos en Jan is verdrietig" hangt af van de waarheidswaarden van "Jan is boos" en "Jan is verdrietig". En in logisch jargon heet het: het voegwoord *en* is waarheidsfunctioneel. Zoals we in § 3.2 hebben gezien betekent "een functie zijn van" hetzelfde als "afhangen van". Vergelijk ook de volgende tweetalige mededeling in een Belgisch restaurant:

> Les prix des plats sont en fonction des prix au marché.
> De prijzen der schotels zijn in functie van de marktprijzen.

Lang niet alle voegwoorden uit het Nederlands zijn waarheidsfunctioneel. Kijk bij voorbeeld naar de volgende zin: "Jan is verdrietig omdat Marie Piet heeft gezoend". Stel dat je weet dat "Jan is verdrietig" en "Marie heeft Piet gezoend" allebei waar zijn. Dan valt nog steeds niet uit de maken of ook de samengestelde zin waar is.

In de propositielogica, afgekort pL, kijken we wat de betekenis van zinnen betreft niet verder dan alleen naar het waar of onwaar zijn van die zinnen. Dit houdt in dat we alleen waarheidsfunctionele voegwoorden kunnen behandelen. Alleen bij die voegwoorden kan de betekenis worden uitgelegd in termen van de betekenissen van de zinnen die worden samengevoegd tot een nieuwe zin. De behandelde voegwoorden zijn:

– ... en ...
– ... of ...
– als ... dan ...
– ... desda[1] ...
– niet ...

[1] 'desda' is jargon voor 'dan, en slechts dan, als'; zie sectie 2.4 op blz. 23

Het laatste voegwoord uit het rijtje is er een dat gecombineerd met een enkele zin al een nieuwe zin oplevert. Onze waarheidsfunctionele kijk betekent dat we afzien van allerlei aspecten in de betekenis van deze woorden. Toch blijken er ondanks deze beperkingen nog aardig wat redeneringen met behulp van propositielogica te analyseren.

7.2 Waarheidstafels

Alles wat we in de propositielogica nodig hebben zijn letters voor beweringen (*propositieletters*; ook wel: *propositionele variabelen* of *propositie-variabelen*) en symbolen voor de voegwoorden. De voegwoorden noemen we voortaan *logische connectieven*. Het connectief voor negatie, $\neg$, combineert met één bewering, en we noemen het daarom *eenplaatsig*. Stel dat p een propositieletter is. Dan is $\neg p$ een samengestelde bewering, met als betekenis: "het is niet zo dat p". Het doet er niet toe voor welke bewering p staat: daarvan hadden we nu juist geabstraheerd. Het symbool voor *en* is $\wedge$ (ook veel gebruikt wordt &). We noemen $\wedge$ het conjunctieteken. $\vee$ staat voor *of* (Latijn: *v*el); we noemen dit het disjunctieteken. $\rightarrow$ staat voor *als dan*: het implicatieteken. Tenslotte hebben we $\leftrightarrow$ voor *desda*: het equivalentieteken. De connectieven $\wedge$, $\vee$, $\rightarrow$, en $\leftrightarrow$ combineren twee zinnen tot een nieuwe zin: ze heten daarom *tweeplaatsige connectieven*.

Zoals gezegd gaan we de connectieven waarheidsfunctioneel opvatten. Een mooie manier om de betekenis van de connectieven weer te geven is in de vorm van een zogenaamde *waarheidstafel*: een tabel waarin is uitgespeld welke waarheidswaarde het connectief oplevert voor alle mogelijke combinaties van waarheidswaarden van de zinnen die met behulp van dat connectief zijn samengevoegd.

We gebruiken φ en ψ voor willekeurige propositielogische beweringen. In jargon: deze symbolen zijn *metavariabelen* die staan voor propositielogische beweringen. Een precieze definitie van 'propositielogische beweringen' volgt in § 7.4; voorlopig: alle beweringen die met behulp van de bovengenoemde connectieven kunnen worden gevormd. Een waarheidstafel voor $\wedge$ ziet er nu als volgt uit:

φ	ψ	$(\varphi \wedge \psi)$
waar	waar	waar
onwaar	waar	onwaar
waar	onwaar	onwaar
onwaar	onwaar	onwaar

Een formule van de vorm $(\varphi \wedge \psi)$ heet een *conjunctie*. De twee onderdelen φ en ψ noemen we de *conjuncten*.

Korter opschrijven van de waarheidstafel voor $\wedge$ kan ook. We gebruiken 1 voor *waar* en 0 voor *onwaar*. Het volgende staatje geeft dezelfde waarheidstafel in een andere vorm:

$\wedge$	1	0
1	1	0
0	0	0

Vanaf nu zullen we steeds 1 en 0 gebruiken voor respectievelijk *waar* en *onwaar*. We zeggen: de verzameling van waarheidswaarden is de verzameling $\{0,1\}$.

Even terzijde: het *en* uit de omgangstaal kan natuurlijk niet helemaal in zo'n tabel gevangen worden. Immers, indien dat zo zou zijn, dan zouden de volgende twee zinnen hetzelfde moeten betekenen, *quod non*:

Hij slaakte een diepe zucht en hij overleed.

Hij overleed en hij slaakte een diepe zucht.

In (7.2) vindt de zucht voor het sterven plaats. In de propositielogica hebben we daarentegen: $(\varphi \wedge \psi)$ is waar desda $(\psi \wedge \varphi)$ waar is. In jargon: het waarheidsfunctionele connectief $\wedge$ is *commutatief*.

Een waarheidstafel voor negatie ziet er zo uit (we geven beide notatiewijzen naast elkaar):

φ	$\neg\varphi$
1	0
0	1

	$\neg$
1	0
0	1

Je ziet het: wat $\neg$ doet is de twee waarheidswaarden omdraaien.

Bij *of* moeten we oppassen: de natuurlijke taal kent zowel het inclusieve als het exclusieve *of*. Exclusief *of* wordt meestal aangegeven door *of*...*of*...: "Luister eens Jantje: of je krijgt een ijsje, of een glas limonade (maar niet zeuren om allebei)". Het disjunctieteken $\vee$ staat voor het inclusieve *of*. Hier is de waarheidstafel:

$\vee$	1	0
1	1	1
0	1	0

Opgave 7.1 Geef zelf een waarheidstafel voor het exclusieve *of* (gebruik het symbool a afgeleid van het Latijnse *aut*).

Een formule van de vorm $(\varphi \vee \psi)$ heet een *disjunctie*, en φ en ψ zijn hierin de *disjuncten*.

Een formule van de vorm $(\varphi \to \psi)$ heet een *materiële implicatie* of kortweg *implicatie*; φ is hierin de *voorzin* en ψ de *nazin*. De voorzin in een implicatie wordt ook wel de *antecedent* genoemd, en de nazin de *consequent*.

De waarheidstafel voor $\to$ is vaak een didactisch struikelblok:

$\to$	1	0
1	1	0
0	1	1

Het begripsprobleem hier is dat het *als dan* uit de natuurlijke taal niet waarheidsfunctioneel is. We verwachten in een *als dan* zin dat er een of ander inhoudelijk verband is tussen de voor- en de nazin. Om een voorbeeld te noemen: "Als Marie Piet zoent, dan wordt Jan boos" wordt meestal zo opgevat dat Jan's boosheid iets met dat gezoen te maken heeft. Jan wordt boos *omdat* er gezoend wordt. Dit verband is echter niet waarheidsfunctioneel, dus in de propositielogica kan het niet worden uitgedrukt.

De achtergrond van de waarheidstafel voor $\to$ wordt gevormd door de wens om $\to$ te gebruiken voor de analyse van *als* ... *dan* ... in wiskundig spraakgebruik. Zie de volgende voorbeeld-bewering.

Als een verzameling A eindig is, dan is A hoogstens aftelbaar.

Bewering (7.2) is van de vorm $(\varphi \to \psi)$, en de bewering is waar. Omdat de bewering van toepassing is op verzamelingen van willekeurige grootte kunnen we het volgende zeggen:

1. Wanneer de antecedent en de consequent allebei waar zijn (het geval waar A een eindige verzameling is) is implicatie (7.2) waar.

2. Wanneer de antecedent en de consequent allebei onwaar zijn (het geval waar A een overaftelbare verzameling is) is implicatie (7.2) waar.

3. Wanneer de antecedent onwaar is en de consequent waar (het geval waar A een aftelbare verzameling is) is implicatie (7.2) waar.

4. De enige manier om (7.2) onwaar te maken is door een verzameling A te vinden die eindig is zonder hoogstens aftelbaar te zijn, dat wil zeggen zonder dat er een bi-jectie tussen A en (een beginstuk van) $\mathbb{N}$ bestaat. Tegelijk voldoen aan beide eisen is per definitie onmogelijk. Dit vierde geval illustreert dat een implicatie onwaar is wanneer de antecedent waar is maar de consequent onwaar.

Deze overwegingen geven aanleiding tot de bovenstaande waarheidstabel. De waarheidsfunctionele manier om de knoop door te hakken wat betreft de betekenis van *als* ... *dan* ... heeft echter tot gevolg dat we de volgende twee zinnen—als we ze propositielogisch analyseren—waar moeten noemen:

Als 5 even is, dan is Beatrix koningin van Nederland.

Als 5 even is, dan is Beatrix geen koningin van Nederland.

De voorbeeldzin die nu volgt, daarentegen, is onwaar:

Als 5 oneven is, dan was Beatrix in 1989 geen koningin van Nederland.

Ten overvloede wellicht (maar het is wel belangrijk): voor de analyse van implicatie in natuurlijke taal zitten we niet aan de propositielogica vastgebakken; er bestaat een keur van voorstellen om zwaarder logisch geschut in

stelling te brengen. Wij houden ons echter hier nog even bezig met het *aap noot mies* van de logica: het waarheidsfunctioneel analyseren van beweringen.

Tenslotte een tafel voor de equivalentie:

$\leftrightarrow$	1	0
1	1	0
0	0	1

Deze waarheidsfunctionele equivalentie wordt ook wel *materiële equivalentie* genoemd.
We hebben al gezien dat het tweeplaatsige connectief $\wedge$ commutatief is.

Opgave 7.2 Ga aan de hand van de waarheidstafels na welke tweeplaatsige connectieven commutatief zijn en welke niet.

Met behulp van de waarheidstafels die we hierboven gegeven hebben kunnen we nu waarheidstafels gaan maken voor willekeurig complexe formules. De precieze regels voor het construeren van complexe formules vind je in § 7.4, maar we lopen alvast even vooruit:
$((p \rightarrow q) \wedge (q \rightarrow p))$ is een complexe formule. We zullen voor deze formule een waarheidstafel gaan construeren. De constructie bestaat uit het stap voor stap uitrekenen van de waarheidswaarden voor de deelformules, met behulp van de informatie uit de waarheidstafels voor de verschillende connectieven:

p	q	$(p \rightarrow q)$	$(q \rightarrow p)$	$((p \rightarrow q) \wedge (q \rightarrow p))$
1	1	1	1	1
0	1	1	0	0
1	0	0	1	0
0	0	1	1	1

Deze waarheidstafel toont aan dat de volgende twee formules 'op hetzelfde neerkomen' (vergelijk de tafel voor $\leftrightarrow$):

– $((p \rightarrow q) \wedge (q \rightarrow p))$

– $(p \leftrightarrow q)$.

In termen die we in § 8 nader zullen toelichten: deze twee formules zijn logisch equivalent.

Opgave 7.3 Geef een waarheidstafel voor $\neg(p \leftrightarrow q)$. Geef vervolgens een andere formule die dezelfde waarheidstafel heeft (dat wil zeggen: die logisch equivalent is).

Propositielogica speelt in vele programmeertalen een grote rol. Propositielogische uitdrukkingen heten daar veelal *Boolese uitdrukkingen* (Eng.: *Booleans*), naar de logicus George Boole (1815–1864). Boolese uitdrukkingen kunnen worden gecombineerd met behulp van propositielogische voegwoorden `AND`, `OR`, `NOT`. Het is verleidelijk ook de bekende `IF THEN` en `IF THEN ELSE` constructies op te vatten als propositielogische voegwoorden, maar dit is strikt genomen niet correct, want het volgende stukje imperatieve code is geen Boolese uitdrukking maar een imperatieve opdracht:

```
IF geslaagd THEN write('ok') ELSE write('fout')
```

In deze imperatieve opdracht is `geslaagd` een Boolese variabele, dat wil zeggen een atomaire Boolese uitdrukking; `write('ok')` en `write('fout')` zijn imperatieve opdrachten. Boolese variabelen zijn programmeertaal-tegenhangers van propositieletters; het zijn atomaire uitdrukkingen die de twee waarden *waar* en *onwaar* kunnen aannemen. De bovenstaande opdracht is equivalent met het volgende tweetal opdrachten:

```
IF geslaagd THEN write('ok');
IF NOT geslaagd THEN write('fout')
```

In de tweede opdracht van dit paar is `NOT geslaagd` een voorbeeld van een samengestelde Boolese uitdrukking.

7.3 BNF regels

De formele talen waarmee we ons in de logica en informatica bezighouden kunnen worden beschreven met behulp van zogenaamde *contextvrije herschrijfregels* of *contextvrije productieregels*. We zullen hier de notatie voor contextvrije herschrijfregels volgen die in informatica-kringen gebruikelijk is. Dergelijke herschrijfregels worden door informatici *BNF regels* genoemd. BNF staat voor *Backus-Naur Form*; Backus en Naur zijn twee informatici die het gebruik van dit soort herschrijfregels voor de formele definitie van programmeertalen hebben gepropageerd.

Hier is een voorbeeld van een omschrijving van een heel eenvoudig taalfragment met behulp van contextvrije herschrijfregels.

Z ::= $A\ B$ **.**
A ::= $C\ D$
B ::= $E\ A$
C ::= **elke** | **een** | **geen**
D ::= **man** | **vrouw**
E ::= **bemint** | **haat**

Een verzameling herschrijfregels zoals deze noemen we een *herschrijfgrammatica*. Elke herschrijfregel heeft een linkerkant en een rechterkant, die van elkaar worden gescheiden door het speciale symbool *::=*. De linkerkant bestaat steeds uit een enkel symbool. De rechterkant bestaat uit een rijtje van symbolen; we vatten daarbij elk van de vetgedrukte woorden als een enkel symbool op. In sommige van de rijtjes aan de rechterkant van het herschrijfsymbool ::= komt het speciale symbool | voor: dit symbool dient om alternatieven aan te geven. Beschouw de volgende regel:

D ::= **man** | **vrouw**

Dit betekent dat symbool D kan worden herschreven als het woord *man* of als het woord *vrouw*. De vetgedrukte woorden in de laatste drie regels van de voorbeeldgrammatica en de vetgedrukte punt in de eerste regel zijn *eindsymbolen*. De symbolen Z, A, B, C, D en E zijn *hulpsymbolen*.

Bij elke herschrijfgrammatica hoort een afspraak over het symbool waarmee het herschrijven begint. De bovenstaande voorbeeldgrammatica is bedoeld om Nederlandse zinnen op te leveren. Om zo'n Nederlandse zin te krijgen moeten we beginnen met het herschrijven van het symbool *Z*. Dit symbool heet het *beginsymbool* of het *startsymbool*.

Om na te gaan welke rijtjes van eindsymbolen zinnen van de door de herschrijfgrammatica beschreven taal zijn, starten we met het beginsymbool, en we gaan dat door toepassing van een productieregel *herschrijven* tot een nieuw symboolrijtje; op het resultaat passen we opnieuw een productieregel toe, enzovoort: we vervangen steeds een symbool dat in de grammatica links van een pijl voorkomt door wat er rechts van die pijl staat. Hier is een voorbeeld van dit proces:

> *Z* ⇒ *A B* **.** ⇒ *C D B* **.** ⇒ **elke** *D B* **.** ⇒ **elke man** *B* **.** ⇒
> **elke man** *E A* **.** ⇒ **elke man bemint** *A* **.** ⇒
> **elke man bemint** *C D* **.** ⇒ **elke man bemint een** *D* **.** ⇒
> **elke man bemint een vrouw .**

Een keten van herschrijvingen als die uit het voorbeeld noemen we een *afleiding*. Het voorbeeld toont dus aan dat in onze grammatica het startsymbool *Z* herschreven kan worden tot *elke man bemint een vrouw.*. Anders gezegd: er is een afleiding van *elke man bemint een vrouw.* uit *Z*. Een afleiding uit *Z* van een rijtje dat alleen bestaat uit eindsymbolen noemen we een *zin*. In plaats van

> Er bestaat een afleiding van *elke man bemint een vrouw.* uit *Z*

zegt men ook wel:

> *Z* brengt *elke man bemint een vrouw.* voort,

of:

> *Z* genereert *elke man bemint een vrouw.*.

Dus: *elke man bemint een vrouw.* is een *zin* die door onze voorbeeldgrammatica wordt voortgebracht.

De pijl ⇒ geeft aan dat het symboolrijtje dat rechts van de pijl staat *direct* kan worden afgeleid van het symboolrijtje links ervan. Aan een directe afleiding komt slechts één productieregel te pas. Een afleiding waar meer dan een productieregel aan te pas komt heet *indirect*. We kunnen dus zeggen: er bestaat een indirecte afleiding van *elke man bemint een vrouw.* uit *Z*. Onze voorbeeldgrammatica genereert de volgende verzameling zinnen:

- een man bemint een man.
- elke man bemint een man.
- geen man bemint een man.
- een vrouw bemint een man.
- elke vrouw bemint een man.
- geen vrouw bemint een man.
- een man bemint een vrouw.
- ...

Opgave 7.4 Hoeveel verschillende zinnen genereert deze grammatica?

Opgave 7.5 Ga na waarom de volgende voorbeeldgrammatica oneindig veel zinnen genereert:

> Z ::= **a** Z | **b**

Probeer een algemeen antwoord te geven op de vraag onder elke voorwaarde een eindige verzameling BNF regels oneindig veel zinnen genereert.

We resumeren nog even de verschillende soorten van symbolen die in de BNF herschrijfregels te onderscheiden vallen:

1. **metasymbolen:** Dit zijn de symbolen die worden gebruikt om aan te geven *hoe* de BNF regel moet worden gelezen. Ze hebben niets te maken met de taal die wordt gedefinieerd. Het zijn de symbolen ::= en | . We beschouwen ::= als een enkel symbool.

2. **hulpsymbolen:** Dit zijn symbolen die met behulp van BNF regels worden herschreven, maar die zelf niet in de gedefinieerde taal voorkomen. Wij zullen hulpsymbolen steeds *cursief* schrijven. Een van de hulpsymbolen neemt een speciale plaats in: het startsymbool.

3. **eindsymbolen:** Dit zijn de symbolen die tot de gedefinieerde taal behoren. Eindsymbolen worden zelf nooit herschreven; ze kunnen alleen aan de rechterkant van BNF regels voor. Wij zullen eindsymbolen steeds **vet** schrijven.

Hulpsymbolen en eindsymbolen kunnen uit meerdere tekens bestaan; we beschouwen het woord *man* als een enkel eindsymbool. Wanneer we ook voor de hulpsymbolen woorden gebruiken kunnen we de structuur van het fragment verduidelijken:

zin ::= *nominale-constituent verbale-constituent* **.**
nominale-constituent ::= *determinator nomen*
verbale-constituent ::= *werkwoord nominale-constituent*
determinator ::= **elke** | **een** | **geen**
nomen ::= **man** | **vrouw**
werkwoord ::= **bemint** | **haat**

Soms worden er naast ::= en | nog andere metasymbolen gebruikt. We gaan daar hier niet op in, want voor onze eerste toepassingen bij het definiëren van logische talen weet je nu genoeg.

7.4 De syntaxis van de propositielogica

In wiskunde en logica wordt veel gewerkt met een manier van definiëren die we *inductief* of *recursief* noemen. Het recursief definiëren van een verzameling A gaat zo.

– We noemen eerst een eindig aantal dingen waarvan we meedelen dat ze elementen zijn van de verzameling A. Dit is de *basisclausule* van de definitie.

– Vervolgens zeggen we: als we een ding hebben dat in verzameling A zit, en we voeren daar een bepaalde bewerking op uit, dan is het resultaat weer een element van A. Dit is de *recursie-clausule* van de definitie.

– Tenslotte zeggen we: behalve de elementen die je op de bovengenoemde manieren in een eindig aantal stappen kunt vormen heeft A geen elementen. Dit heet: de *afsluitingsclausule* van de recursieve definitie.

Een verzameling A kan ook recursief worden gedefinieerd zonder afsluitingsclausule. De formulering wordt dan:

– A is de *kleinste* verzameling zo dat

1. de basisclausule opgaat,
2. de recursieve clausule opgaat.

Deze formulering komt precies op hetzelfde neer.

We geven een aantal voorbeelden van recursieve definities: Eerst een recursieve definitie van *mens*. Daar gaat ie.

– Ten eerste: Adam en Eva zijn mensen.

– Ten tweede: kinderen van mensen zijn mensen.

– Tenslotte: verder zijn er geen mensen.

Gegeven deze definitie kun je nu gaan controleren of een bepaald object een mens is. Laten we prins Bernhard nemen. Is prins Bernhard een mens? Die vraag kunnen we herleiden tot de vraag of de ouders van de prins mensen waren. Nu, dat is niet meteen duidelijk. We moeten kijken naar *hun* ouders. Enzovoorts. Hetzij we komen bij Adam en Eva terecht (let wel: voor de prins zelf en voor alle voorouders van de prins), en Z.K.H. blijkt een mens, hetzij dat gebeurt niet, en de prins valt door de mand (volgens deze definitie).

De definitie van *natuurlijk getal* in de wiskunde gaat net zo. Ten eerste: 0 is een natuurlijk getal. Ten tweede: als iets een natuurlijk getal is, dan is het getal dat je krijgt door 1 bij dat getal op te tellen (of met andere woorden: door de *opvolger*-functie op dat getal los te laten) ook een natuurlijk getal. Tenslotte: niets anders is een natuurlijk getal. Dit levert de bekende verzameling $\mathbb{N} = \{1, 2, 3, 4, \ldots\}$.

In § 2.9 hebben we het gehad over geordende rijtjes van elementen. We hebben toen uitgelegd wat een geordend paar is, en vervolgens gezegd: geordende rijtjes kunnen ook meer dan twee elementen hebben. Naar formeel-wiskundige maatstaven gemeten is dit eigenlijk te vaag. Formeel gesproken dient het begrip 'geordend rijtje' expliciet te worden gedefinieerd, en dat kan gebeuren met een recursieve definitie. Als voorbeeld zullen we hier het begrip *geordend rijtje schrijftekens* of kortweg *tekenrijtje* recursief definiëren. In programmeertalen spelen tekenrijtjes (Eng.: *strings*) een grote rol.

Eerst definiëren we de verzameling schrijftekens (Eng.: *characters*). Dat gaat door opsomming, en we kunnen er een BNF regel voor gebruiken:

schrijfteken ::= **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** |
J | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** |
S | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |
a | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** |
j | **k** | **l** | **m** | **n** | **o** | **p** | **q** | **r** |
s | **t** | **u** | **v** | **w** | **x** | **y** | **z** |
␣ | **,** | **.** | **?** | **!** | **:** | **;**

Hier staat ␣ voor het spatieteken. De recursieve definitie van *tekenrijtje* gaat nu gewoon met een tweetal BNF regels:

tekenrijtje ::= *leeg* | *schrijfteken tekenrijtje*
leeg ::=

Dat we te maken hebben met een recursieve definitie zien we aan het feit dat het hulpsymbool voor *tekenrijtje* zowel aan de linkerkant als de rechterkant in de regel voorkomt. Ga na dat de twee BNF regels samen neerkomen op het volgende:

Definitie 7.1 Tekenrijtjes*:*

– *Een rijtje dat helemaal niets bevat is een tekenrijtje.*

– *Een schrijfteken gevolgd door een tekenrijtje is een tekenrijtje.*

– *Niets anders is een tekenrijtje.*

Je kunt zelf controleren dat *prins Bernhard* een tekenrijtje is.

Nu gaan we de verzameling formules van de propositielogica recursief definiëren. Iets preciezer: we gaan definiëren hoe een propositielogische taal $\mathscr{T}$ eruit ziet. Elke keuze van de propositieletters bepaalt een andere taal.

We zouden de volgende keuze voor de verzameling propositieletters kunnen maken: $\{p, q, r\}$. Ander keuzes zijn ook mogelijk. Zolang de verzameling eindig is kunnen de elementen gewoon worden opgesomd. Het is echter ook geen probleem om oneindig veel propositieletters in te voeren. Dat gaat natuurlijk weer gewoon met recursie. Bij voorbeeld: we willen dat de letter p gevolgd door een willekeurig aantal accenttekens een propositieletter is. Dat wil zeggen: p, p''', p'''''''' zijn voorbeelden van propositieletters. De recursieve definitie van de (oneindige) verzameling $\{p, p', p'', p''', p'''', \ldots\}$ van propositieletters gaat als volgt:

propositieletter ::= **p** | *propositieletter* **'**

Dit komt neer op het volgende:

Definitie 7.2 Propositieletters *van $\mathcal{T}$:*

– p *is een propositieletter;*

– *als A een propositieletter is, dan A′ ook;*

– *niets anders is een propositieletter.*

In deze definitie is A weer een *metavariabele*: A staat voor een willekeurige propositieletter die al eerder is geconstrueerd. Propositieletters die op deze manier zijn geconstrueerd zijn voor ons mensen soms moeilijk uit elkaar te houden, maar een computer draait er zijn hand niet voor om. In gevallen waar je maar een paar propositieletters nodig hebt is het handiger om gewoon verschillende letters van het alfabet te nemen. Vaak worden dan de letters p, q, r en s gebruikt.

Elke verzameling propositieletters bepaalt een propositielogische taal $\mathcal{T}$. Een propositielogische taal $\mathcal{T}$ is niets anders dan een verzameling *welgevormde formules* (Eng.: *well-formed formulas—wffs*). We definiëren met behulp van het begrip *propositieletter* het begrip *welgevormde formule* (voor zekere taal $\mathcal{T}$, waarbij de verzameling propositieletters de taal $\mathcal{T}$ bepaalt). Weer geven we twee versies: eerst met behulp van BNF regels, daarna met een expliciete definitie. Hier is de BNF definitie:

formule ::= *propositieletter* |
¬ *formule* |
(*formule* ∧ *formule*) |
(*formule* ∨ *formule*) |
(*formule* → *formule*) |
(*formule* ↔ *formule*)

Welke taal wordt gedefinieerd hangt af van de BNF regel voor *propositieletter*.

We kunnen ditzelfde ook expliciet in een recursieve definitie verwoorden. Daarbij gebruiken we φ en ψ als meta-variabelen voor propositielogische formules.

Definitie 7.3 Welgevormde formules *van $\mathcal{T}$:*

– *Elke propositieletter van $\mathcal{T}$ is een welgevormde formule van $\mathcal{T}$;*

– *als φ een welgevormde formule van $\mathcal{T}$ is, dan $\neg\varphi$ ook; als φ en ψ welgevormde formules van $\mathcal{T}$ zijn, dan $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ en $(\varphi \leftrightarrow \psi)$ ook;*

– *niets anders is een welgevormde formule van $\mathcal{T}$.*

De propositieletters van $\mathcal{T}$ worden ook wel de *atomaire formules van $\mathcal{T}$* of de *atomen van $\mathcal{T}$* genoemd. φ en ψ—de metavariabelen uit de definitie—staan voor willekeurige formules van de taal $\mathcal{T}$ die al geconstrueerd zijn.

Bij recursieve definities horen *inductieve bewijzen*. Om te bewijzen dat alle welgevormde formules van $\mathcal{T}$ zekere eigenschap E hebben lopen we de recursieve definitie langs, en we controleren twee dingen:

1. dat in het basisgeval E aanwezig is;
2. dat de eigenschap E bij het toepassen van de recursieve clausule bewaard blijft.

De eerste controlestap wordt de *basisstap* genoemd; de tweede controlestap heet de *inductiestap*.

Een kinderachtig voorbeeld: je kunt inductief bewijzen dat elke propositieletter uit de verzameling $\{p, p', p'', p''', \ldots\}$, gedefinieerd als boven, begint met de letter p.

Opgave 7.6 Schrijf dit inductieve bewijs op.

Iets minder kinderachtig, maar nog steeds flauw: je kunt inductief bewijzen dat de welgevormde formules van $\mathcal{T}$ evenveel linker- als rechterhaakjes hebben. Het bewijs gaat zo.

Stelling 7.1 *De welgevormde formules van een propositielogische taal $\mathcal{T}$ hebben evenveel linker- als rechterhaakjes.*

Bewijs:

– Basisstap: propositieletters hebben geen haakjes, dus zeker evenveel linker- als rechter.

– Inductiestap: als φ evenveel linker- als rechterhaakjes heeft, dan $\neg\varphi$ ook: er komen immers geen haakjes bij; als zowel φ als ψ evenveel linker- als rechterhaakjes heeft, dan $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ en $(\varphi \leftrightarrow \psi)$ ook: er komt steeds precies één linker- en één rechterhaakje bij. □

Het is van belang grondig vertrouwd te raken met inductieve bewijzen. De volgende opdracht laat een toepassing zien in het redeneren over verzamelingen.

Opgave 7.7 In § 2.8 hebben we zonder bewijs vermeld dat voor eindige verzamelingen A geldt: als n het aantal elementen van A is, dan is 2^n het aantal elementen van 2^A. Ga na waarom dit zo is. Aanwijzing: je kunt dit als volgt nagaan met behulp van inductie:

– Basisstap: Een verzameling van nul elementen heeft een machtsverzameling van $1 = 2^0$ element.

– Inductiestap: Neem aan dat je al weet dat verzameling A met $n-1$ elementen een machtsverzameling van 2^{n-1} elementen heeft. Laat met behulp van dit gegeven zien dat een verzameling van n elementen tweemaal 2^{n-1} elementen heeft (dat wil zeggen: $2 \times 2^{n-1} = 2^n$ elementen).

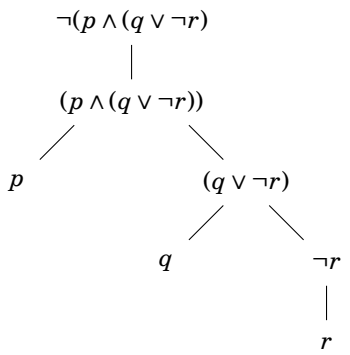
Opgave 7.8 Laat $\mathcal{T}$ de propositielogische taal zijn met als propositieletters de verzameling $\{p, q, r\}$. Scheid de welgevormde formules van $\mathcal{T}$ van de rest:

1. pq
2. $p \wedge q$
3. $(p \wedge q)$
4. $(p \wedge (q \vee r))$
5. (p'')
6. $\neg\neg p$
7. $(\neg\neg p)$
8. $(\neg(\neg p))$
9. $p \vee p$
10. $(p \vee p)$
11. $(p \vee q \vee r)$
12. $(p \rightarrow (p \rightarrow p))$
13. $(p \rightarrow p) \rightarrow p$
14. $\neg\neg\neg\neg\neg\neg\neg q$.

Opgave 7.9 Hoeveel formules heeft de taal $\mathscr{T}$ uit de vorige opdracht?

Opgave 7.10 Als de propositielogische taal $\mathscr{T}$ aftelbaar veel propositieletters heeft, hoeveel formules heeft $\mathscr{T}$ dan?

Je zult bij het maken van opdracht 7.8 hebben gemerkt dat de definitie van *welgevormde formule* heel precies vastlegt waar de haakjes horen. De haakjes zijn van groot belang voor het vermijden van dubbelzinnigheden: ze zorgen ervoor dat elke formule op precies één manier kan worden 'ontleed'. Ontleden van een propositielogische formule is niets anders dan nagaan op welke manier die formule volgens de constructieregels uit de recursieve definitie van *welgevormde formule* is opgebouwd. De opbouw van een formule kunnen we weergeven in een zogenaamde *constructieboom* (Eng.: construction tree). Hier is een voorbeeld, voor de formule $\neg(p \wedge (q \vee \neg r))$:



De *knopen* van de constructieboom worden gevormd door welgevormde formules; de welgevormde formule die de topknoop-positie inneemt is geconstrueerd met behulp van de welgevormde formules op de lager gelegen knopen. Als je wilt kun je voor elke knoop nagaan volgens welke clausule in de recursieve definitie de formule op die knoop is gevormd.

Uit de constructieboom voor een formule kan het zogenaamde *bereik* (Eng.: *scope*) van de verschillende connectieven die in de formule voorkomen worden afgelezen. Een connectief heeft bereik over de formule of formules op de knoop of knopen waar je terecht komt door vanaf de knoop waar het desbetreffende connectief wordt geïntroduceerd langs een tak van de constructieboom precies één stapje naar beneden te doen. In boom-jargon: een connectief heeft bereik over de formules op de knopen die *onmiddellijk worden gedomineerd* door de knoop waar het connectief wordt geïntroduceerd.

Opgave 7.11 Geef van beide negatie-tekens in de formule $\neg(p \wedge (q \vee \neg r))$ het bereik aan.

Opgave 7.12 Maak een constructieboom voor de formule

$$(\neg p \rightarrow (\neg q \wedge r))$$

en geef het bereik van de beide negatie-tekens aan.

Wanneer we in de formule $(p \wedge (q \vee r))$ de haakjes zouden weglaten, zouden we niet meer kunnen nagaan dat deze formule ontstaan is door de conjunctie te nemen van de atomaire formule p en de formule $(q \vee r)$; immers, we zouden dan evengoed te maken kunnen hebben met de disjunctie van de conjunctie van p en q aan de ene kant, en de atomaire formule r aan de andere kant. Door weglating van de haakjes zou de formule $(p \wedge (q \vee r))$ op meerdere manieren kunnen worden gelezen. Omdat deze verschillende lezingen niet op hetzelfde neerkomen zou dit betekenen dat sommige formules van de propositielogica dubbelzinnig zouden zijn geworden. De haakjes dienen om dit te voorkomen. Die haakjes zorgen ervoor dat iedere formule precies één constructieboom heeft.

Er zijn echter ook gevallen waar we het ons kunnen permitteren wat slordiger om te springen met de haakjes. In elke formule die begint met een linkerhaakje en eindigt met een rechterhaakje kunnen die buitenste haakjes worden weggelaten zonder dat dit dubbelzinnigheden oplevert. Dus: in plaats van $(p \wedge (q \vee r))$ kunnen we zonder bezwaar schrijven: $p \wedge (q \vee r)$. Dit weglaten van buitenste haakjes maakt de formules in het algemeen wat leesbaarder: het wordt dan ook veel gedaan.

Opgave 7.13 Geef een stel BNF regels die precies de formules van een propositielogische taal met een gegeven verzameling propositieletters opleveren, maar nu volgens het recept dat buitenhaakjes niet worden geschreven. Aanwijzing: je hebt een extra hulpsymbool (een zg. *naaktloper*) nodig, voor formules zonder buitenhaakjes.

Een andere situatie waarin haakjes zonder gevaar van dubbelzinnigheid kunnen worden weggelaten is in een formule als $(p \wedge (q \wedge (r \wedge s))))$. De volgorde waarin de conjuncties worden genomen doet er voor de betekenis van het geheel niet toe, dus we kunnen net zo goed schrijven: $p \wedge q \wedge r \wedge s$. In het vervolg zullen we ons zo nu en dan dit soort vrijheden veroorloven. Dit betekent niet dat we de definitie van *welgevormde formule* hebben aangepast, maar alleen dat we er een beetje mee sjoemelen waar het geen kwaad kan.

De constructiebomen die we hierboven gegeven hebben zijn familie van de syntactische structuurbomen die in de compilerbouw worden gebruikt. Om de verbanden precies te kunnen bespreken hebben we een formele definitie nodig van het begrip *boom*. Wij geven er echter de voorkeur aan om je eerst intuïtief vertrouwd te laten raken met bomen, en je op die manier te prepareren voor een echt formele behandeling.

We hebben hierboven gezien dat haakjes worden gebruikt om dubbelzinnigheden in propositielogische formules te vermijden. Er bestaat overigens ook een manier om dergelijke dubbelzinnigheden te vermijden zonder gebruik te maken van haakjes, de zogenaamde *prefix notatie* of *Poolse notatie*. De laatste benaming herinnert aan het feit dat deze notatie door Poolse logici is voorgesteld. In de prefix notatie worden de tweeplaatsige connectieven *voor* de formules die ze samenvoegen geplaatst in plaats van *ertussen* zoals bij de gewone notatie. Nog wat jargon: de gewone notatie wordt ook wel *infix* notatie genoemd. Haakjes blijken in de prefix notatie overbodig te zijn. Hier zijn een aantal voorbeelden van Pools genoteerde propositielogische formules:

– p
– $\wedge pq$
– $\wedge \vee pqr$
– $\wedge p \vee qr$
– $\wedge \vee pq \vee r \neg r$.

Opgave 7.14 Geef constructiebomen voor deze formules. Geef ook bij elke formule de corresponderende infix-versie.

De verzameling van welgevormde formules in Poolse notatie (van een taal $\mathscr{T}$ met verzameling propositieletters P) kan weer recursief worden gedefinieerd.

Opgave 7.15 Geef deze definitie.

Opgave 7.16 Geef een stel BNF regels die de verzameling van welgevormde formules van de propositielogica in Poolse notatie oplevert.

Behalve een infix en een prefix notatie voor formules van de propositielogica bestaat er ook een zogenaamde *postfix notatie*, ook wel *omgekeerd Poolse notatie* (Eng.: reverse Polish notation) genoemd. Omgekeerd Poolse notatie is een zeer geschikt voor computerverwerking; een programmeertaal die gebruik maakt van de voordelen van deze notatie is de taal FORTH (zie [Winfield 1983]).

Opgave 7.17 Geef de postfix-versies van de volgende formules in prefix notatie:

1. p
2. $\wedge pq$
3. $\wedge \vee pqr$
4. $\wedge p \vee qr$
5. $\wedge \vee pq \vee r \neg r$
6. $\neg \neg p$.

Opgave 7.18 Geef een recursieve definitie van de verzameling welgevormde formules van een propositielogische taal in postfix notatie (ga uit van een taal $\mathscr{T}$ met P als verzameling propositieletters).

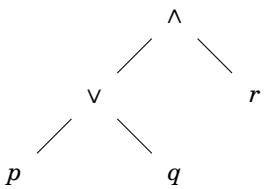
Een laatste notatievariant voor propositielogische formules is de zogenaamde *lijst-notatie* (soms ook Cambridge Pools genoemd, omdat deze notatie gebruikt wordt in LISP, een programmeertaal die is ontwikkeld aan het MIT in Cambridge, Massachusetts). Hier zijn de formules uit opdracht 7.17 in lijstnotatie:

– p
– $(\wedge pq)$
– $(\wedge(\vee pq)r)$
– $(\wedge p(\vee qr))$
– $(\wedge(\vee pq)(\vee r(\neg r)))$
– $(\neg(\neg p))$.

Je ziet het: Pools, maar met kwistig gebruik van extra haakjes.

Opgave 7.19 Geef een recursieve definitie van de verzameling welgevormde formules in lijst-notatie van de propositielogische taal $\mathscr{T}$ met verzameling propositieletters P.

Nauw verwant aan constructiebomen voor formules zijn de zogenaamde *structuurbomen*. Hier is een voorbeeld van een structuurboom voor een formule (let op; we schakelen weer over op de standaardnotatie). De structuurboom voor de formule $((p \vee q) \wedge r)$ is:



Zoals je aan dit voorbeeld kunt zien wordt het bereik van de verschillende connectieven in een structuurboom aangegeven door de positie die die connectieven in de boom innemen. Dus: de rol van de haakjes in een propositielogische formule wordt in een structuurboom voor die formule gespeeld door de hiërarchische verhoudingen tussen de knopen in de boom.

Het is niet zo moeilijk om in te zien dat structuurbomen kunnen dienen als de grondstof waaruit de verschillende notaties voor formules (standaard, Pools, omgekeerd Pools, lijst-notatie) kunnen worden toebereid. Al deze notaties kunnen worden verkregen door mechanische bewerkingen op structuurbomen waarbij de knopen uit die bomen op een bepaalde systematische manier worden verwerkt.

Opgave 7.20 Geef structuurbomen voor de volgende formules:

1. $\neg \neg \neg p$
2. $\neg(p \vee \neg q)$
3. $\neg(\neg p \wedge \neg q)$
4. $((p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p))$.

Hoofdstuk 8

De semantiek van de propositielogica

8.1 Interpretatie van proposities

De semantiek van de propositielogica is een systematische verhandeling over hoe je kunt uitmaken of een propositielogische formule waar is. In die systematische verhandeling wordt royaal gebruik gemaakt van begrippen uit de verzamelingenleer.

Laat P een verzameling propositieletters zijn, en $\mathcal{T}$ de bijbehorende propositielogische taal. De verzameling welgevormde formules van $\mathcal{T}$ noemen we $\mathrm{WF}_{\mathcal{T}}$. We zijn nu geïnteresseerd in zogenaamde *waarderingen* of *valuaties* of *valuatie-functies* (Eng.: valuations) voor $\mathrm{WF}_{\mathcal{T}}$. Een waardering voor de propositielogische taal $\mathcal{T}$ is een functie V die aan alle formules van $\mathcal{T}$ een waarheidswaarde toekent, en die dat doet op een waarheidsfunctionele manier. Iets preciezer: een *waardering* V voor propositielogische taal $\mathcal{T}$ is een functie met domein $\mathrm{WF}_{\mathcal{T}}$ en codomein $\{0,1\}$. V levert voor elke formule van $\mathcal{T}$ een waarheidswaarde op. Wat wil het nu zeggen dat V waarheidsfunctioneel is? Ruwweg het volgende: V houdt zich aan de betekenissen van de connectieven.

We hebben in § 7.1 en § 7.2 al gezien dat de propositielogische connectieven *waarheidsfunctioneel* zijn. Dit houdt in dat twee waarderingen V en V' voor een taal $\mathcal{T}$ alleen verschillend kunnen zijn—dat wil zeggen: voor sommige welgevormde formules verschillende waarden kunnen opleveren—wanneer ze verschillende waarheidswaarden toekennen aan minstens één propositieletter. Immers, gesteld dat V en V' aan elk van de propositieletters van $\mathcal{T}$ dezelfde waarde toekennen, dan worden de toekenningen voor alle andere formules *afgedwongen* door de waarheidstafels. Oftewel: V en V' kennen dan aan elke formule dezelfde waarde toe. Daarmee bedoelen we dat voor elke formule φ geldt:

$$V(\varphi) = V'(\varphi).$$

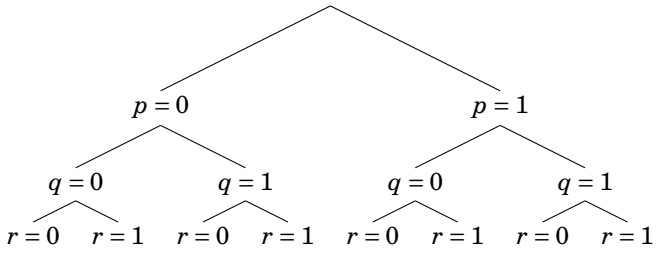
Gevolg van dit alles is: als we geïnteresseerd zijn in valuatie-functies hoeven we alleen te kijken naar de toekenningen van waarheidswaarden aan de propositieletters van onze taal. Met die toekenning ligt de waardering voor *alle* formules van de taal vast. Een toekenning van waarheidswaarden aan de propositieletters van de taal $\mathcal{T}$ heet een *interpretatie* voor $\mathcal{T}$. Formeel:

Definitie 8.1 *Als $\mathcal{T}$ de propositielogische taal is met P als verzameling propositieletters, dan is een functie $I : P \to \{0,1\}$ een* **interpretatie** *voor $\mathcal{T}$.*

Zoals we in § 3.5 hebben gezien heet een functie met de verzameling $\{0,1\}$ als codomein een karakteristieke functie. Karakteristieke functies corresponderen met deelverzamelingen, dus we kunnen zeggen: een interpretatie voor $\mathcal{T}$ karakteriseert een deelverzameling van de verzameling P van propositieletters van $\mathcal{T}$.

Laten we als voorbeeld de taal $\mathcal{T}$ nemen die gebaseerd is op de verzameling $\{p,q,r\}$ van propositieletters. Nu kun je een interpretatie I_1 voor $\mathcal{T}$ geven door te zeggen: $I_1(p) = 1$, $I_1(q) = 1$, $I_1(r) = 0$. Interpretatie I_2 kan er zo uitzien: $I_2(p) = 1$, $I_2(q) = 0$, $I_2(r) = 0$. I_2 verschilt nu van I_1 in de toekenning van een waarde aan de propositieletter q.

Voor een taal met drie propositieletters zijn er $2^3 = 8$ mogelijke interpretaties. Immers, bij elke extra propositieletter verdubbelt het aantal mogelijkheden, zoals uit het volgende plaatje blijkt (stel dat p, q en r de propositieletters zijn):



Uit het plaatje blijkt dat er in totaal 8 mogelijkheden zijn. Dit klopt ook met het deelverzamelingen-perspectief: een verzameling met 3 elementen heeft 8 onderling verschillende deelverzamelingen, een verzameling met n elementen heeft 2^n onderling verschillende deelverzamelingen.

Waarom heet een functie van propositieletters naar waarheidswaarden nu een *interpretatie*? Omdat zo'n functie de rol vervult van 'een specifieke betekenis geven' aan de propositieletters. Wat een interpretatie-functie doet komt op hetzelfde neer als het vervangen van de verzameling propositieletters door een verzameling direct controleerbare beweringen over de werkelijkheid.

Een interpretatie in de huis-, tuin- en keukenzin van een propositielogische taal met als propositieletters de verzameling {p, q, r} zou bij voorbeeld kunnen zijn:

$p \to$ 'Het regent op koninginnedag '88 in Soestdijk.'
$q \to$ 'Het sneeuwt op koninginnedag '88 in Soestdijk.'
$r \to$ 'Het hagelt op koninginnedag '88 in Soestdijk.'

Helaas, dit is nogal een mond vol. En we hebben gezien dat het enige onderscheid dat de propositielogica kan

maken is: dat tussen ware beweringen en onware. Elke ware bewering is even goed als een andere ware bewering. Dit betekent dat we—in plaats van het bovenstaande—net zo goed kunnen zeggen:

$$\begin{array}{lcl} p & \longrightarrow & \text{een willekeurige ware bewering} \\ q & \longrightarrow & \text{een willekeurige onware bewering} \\ r & \longrightarrow & \text{een willekeurige onware bewering} \end{array}$$

Maar dat is weer niets anders dan de volgende interpretatiefunctie met domein $\{p,q,r\}$:

$$\begin{array}{lcl} p & \longrightarrow & 1 \\ q & \longrightarrow & 0 \\ r & \longrightarrow & 0 \end{array}$$

Vandaar dat de definitie van 'interpretatie' is zoals hij is.

We gaan het nu hebben over de manier waarop een interpretatiefunctie I kan worden uitgebreid tot een waardering V_I voor alle welgevormde formules van de taal. Let op: het uitbreidingsvoorschrift volgt de inductieve definitie van de welgevormde formules. We nemen als voorbeeld de taal $\mathscr{T}$ met de verzameling propositieletters $\{p,q,r\}$. Laat een interpretatiefunctie I voor deze verzameling gegeven zijn. We definiëren nu eerst V_I voor atomaire formules, daarna voor willekeurige formules.

– $V_I(p) = I(p)$; $V_I(q) = I(q)$; $V_I(r) = I(r)$.

In het algemeen: als φ een atomaire formule is, dan stellen we:

– $V_I(\varphi) = I(\varphi)$.

De functie I was gedefinieerd voor alle atomaire formules. V_I is voor atomaire formules gewoon gelijk aan I.

– $V_I(\neg\varphi) = 1$ desda $V_I(\varphi) = 0$.

Toelichting: Gesteld dat we al weten welke waarde V_I toekent aan φ, dan kunnen we met behulp van deze clausule de waarde voor $\neg\varphi$ afleiden.

– $V_I((\varphi \wedge \psi)) = 1$ desda $V_I(\varphi) = 1$ en $V_I(\psi) = 1$.

Toelichting: Gesteld dat we de waarheidswaarden van φ en ψ al weten, dan kunnen we die van $(\varphi \wedge \psi)$ afleiden. Hetzelfde geldt voor de samenstellingen die met behulp van de andere connectieven kunnen worden gevormd, zoals blijkt uit de volgende clausules.

– $V_I((\varphi \vee \psi)) = 1$ desda $V_I(\varphi) = 1$ of $V_I(\psi) = 1$.

– $V_I((\varphi \rightarrow \psi)) = 0$ desda $V_I(\varphi) = 1$ en $V_I(\psi) = 0$.

– $V_I((\varphi \leftrightarrow \psi)) = 1$ desda $V_I(\varphi) = V_I(\psi)$.

Hiermee hebben we alle gevallen gehad, en dus hebben we V_I nu inductief gedefinieerd voor alle formules van $\mathscr{T}$. Aan de hand van de waarheidstafels voor de verschillende connectieven kun je nagaan dat de bovenstaande clausules juist zijn.

Nu we beschikken over valuaties kunnen we waarheidstafels in een nieuw licht gaan bekijken. Een waarheidstafel is niets anders dan een constructie van alle mogelijke valuaties voor een formule, uitgaande van de verschillende interpretatiemogelijkheden van de propositieletters die erin voorkomen.

Als we een tafel willen maken voor de formule $\neg(\neg p \vee \neg q)$, dan beginnen we met alle mogelijke interpretaties voor de twee propositieletters p en q. Dat zijn er vier:

	p	q
I_1	1	1
I_2	0	1
I_3	1	0
I_4	0	0

Het opbouwen van de waarderingen V_1, V_2, V_3 en V_4 gaat nu als volgt:

	p	q	$\neg p$	$\neg q$	$(\neg p \vee \neg q)$	$\neg(\neg p \vee \neg q)$
V_1	1	1	0	0	0	1
V_2	0	1	1	0	1	0
V_3	1	0	0	1	1	0
V_4	0	0	1	1	1	0

Merk op dat uit de waarheidstafel blijkt dat $\neg(\neg p \vee \neg q)$ op hetzelfde neerkomt als $(p \wedge q)$.

We zijn weer toe aan het invoeren van wat nieuw logisch jargon.

Definitie 8.2 *Twee propositielogische formules φ en ψ heten* **logisch equivalent** *als voor elke mogelijke waardering V geldt: $V(\varphi) = V(\psi)$.*

In waarheidstafel-termen betekent dit: φ en ψ zijn logisch equivalent als ze dezelfde waarheidstafel opleveren. We kunnen nu dus zeggen: de formules $\neg(\neg p \vee \neg q)$ en $(p \wedge q)$ zijn logisch equivalent.

Opgave 8.1 Ga na dat $\neg(p \wedge q)$ en $(\neg p \vee \neg q)$ logisch equivalent zijn.

Definitie 8.3 *Een formule φ heet een* **tautologie** *(of: een* **logische waarheid***, of ook:* **geldig***) als voor elke mogelijke waardering V geldt: $V(\varphi) = 1$.*

In waarheidstafel-termen: φ is een tautologie als de waarheidstafel voor φ louter enen heeft.

Opgave 8.2 Ga met behulp van waarheidstafels na dat de volgende formules tautologieën zijn (voor elke keuze van φ, ψ en χ ; we hebben hier dus eigenlijk te maken met tautologie-*schema's* of tautologie-sjablonen):

1. $\varphi \vee \neg\varphi$ ['wet van de uitgesloten derde']
2. $(\varphi \wedge \psi) \rightarrow \varphi$
3. $\varphi \rightarrow (\varphi \vee \psi)$
4. $\neg\varphi \rightarrow (\varphi \rightarrow \psi)$ ['ex falso sequitur quodlibet']
5. $((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi$ ['wet van Peirce']
6. $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$.

Definitie 8.4 *Een formule φ heet een* **contradictie** *(of: een* **logische onwaarheid***) als voor elke mogelijke waardering V geldt: $V(\varphi) = 0$.*

In waarheidstafel-termen: φ is een contradictie als de waarheidstafel voor φ louter nullen heeft.

Definitie 8.5 *Een formule φ heet* **vervulbaar** *als er een waardering V bestaat waarvoor geldt: $V(\varphi) = 1$.*

Een formule is vervulbaar als het geen contradictie is.

Opgave 8.3 Welke van de volgende formules zijn tautologieën? En welke zijn contradicties?

1. $(p \rightarrow p)$
2. $\neg(p \rightarrow p)$
3. $(p \rightarrow (\neg p \rightarrow p))$
4. $((p \wedge q) \leftrightarrow \neg(\neg p \vee \neg q))$
5. $(\neg p \rightarrow (p \rightarrow (q \rightarrow r)))$
6. $(\neg(p \leftrightarrow q) \rightarrow ((p \leftrightarrow r) \vee (q \leftrightarrow r)))$.

Opgave 8.4 Laat zien dat de ontkenning van een tautologie altijd een contradictie is en omgekeerd.

Opgave 8.5 Geef een voorbeeld van een formule die noch een tautologie, noch een contradictie is.

De laatste opdracht was gemakkelijk: je zult eruit begrepen hebben dat lang niet alle formules tautologieën of contradicties zijn.

Definitie 8.6 *Een formule die noch een tautologie, noch een contradictie is noemen we een* **contingente** *formule.*

We vertalen het bovenstaande jargon weer even in huis-, tuin- en keukentermen. De waarheid van een tautologie hangt niet af van de interpretatie van de propositieletters die erin voorkomen. Dat maakt hem nu juist *logisch* waar: toetsing aan de werkelijkheid, kennis van de wereld, of iets dergelijks, is *niet* nodig om de waarheid van een tautologie vast te stellen. Nog anders gezegd: ik hoef niet uit het raam te kijken om vast te stellen dat de bewering 'het regent of het regent niet' waar is. Een parafrase van 'contingente formule': contingente formules zijn formules waarvan de waarheidswaarde afhangt van de interpretatie. Nog anders: contingente formules hebben een waarheidstafel met *niet* alleen enen en *niet* alleen nullen.

We zijn nu toe aan de definitie van *logisch gevolg*, het kernbegrip uit de propositielogica. Dit is het belangrijkste stukje logisch jargon uit de propositielogica omdat propositielogica uiteindelijk de leer van het correct propositielogisch redeneren is (vergelijk ook hoofdstuk 4).

Definitie 8.7 *Een formule φ heet een* **logisch gevolg** *van formule ψ als voor elke valuatie V met $V(\psi) = 1$ geldt dat $V(\varphi) = 1$.*

Net zo voor het geval waar er meerdere premissen zijn, zeg $\psi_1, \ldots, \psi_n$. We zeggen dan: de premissenverzameling is de verzameling $\{\psi_1, \ldots, \psi_n\}$. Noem de premissenverzameling Σ. Dan kunnen we de definitie van *logisch gevolg* generaliseren:

Definitie 8.8 *Formule φ is een* **logisch gevolg** *van de premissenverzameling Σ als voor elke valuatie V die voor elke formule in Σ de waarde 1 oplevert geldt: $V(\varphi) = 1$.*

We noteren dit als:

$$\psi_1, \ldots, \psi_n \models \varphi$$

of ook wel als:

$$\{\psi_1, \ldots, \psi_n\} \models \varphi$$

of gewoon:

$$\Sigma \models \varphi.$$

Hier zijn $\psi_1, ..., \psi_n$ de premissen, en is φ de conclusie. We zeggen ook wel: het redeneerpatroon 'concludeer uit premissenverzameling Σ tot conclusie φ' is *geldig*. We zijn hier in feite weer terug bij ons uitgangspunt uit hoofdstuk 4: we hebben nu een exacte definitie gegeven van het begrip *logische gevolgtrekking* (of: *geldige redenering*) voor de taal van de propositielogica.

Het symbool $\models$ wordt overigens vanwege zijn vorm vaak aangeduid als *turnstile* (je weet wel, dat draaipoortje waar je doorheen moet als je de supermarkt in wil). Omdat er ook een enkele turnstile, $\vdash$, bestaat, wordt $\models$ ook wel aangeduid als *dubbele turnstile*.

De bovenstaande definitie van het begrip *logisch gevolg* maakt het mogelijk nog een iets andere parafrase te geven van het begrip *tautologie*. Een tautologie is een logisch gevolg van de lege premissenverzameling. Ga na dat deze parafrase klopt. Als φ een tautologie is, dan kunnen we dat zo opschrijven:

$$\models \varphi.$$

Immers, φ volgt dan logisch uit de lege premissenverzameling. 'φ is geen tautologie' noteren we wel als:

$$\not\models \varphi.$$

We zeggen in dit geval ook wel: formule φ is niet logisch geldig. Let op: $\not\models \varphi$ houdt nog niet in dat φ een contingente formule is. Contingentie wil zeggen:

$$\not\models \varphi \text{ en } \not\models \neg\varphi.$$

We kunnen nu nog een keer met andere woorden zeggen wat het betekent dat twee formules φ en ψ logisch equivalent zijn: φ en ψ heten *logisch equivalent* als ze logisch gevolg zijn van elkaar, dat wil zeggen: als elke valuatie die φ waar maakt ook ψ waar maakt, en omgekeerd.

Gewoontegetrouw stappen we tot slot nog even over naar het waarheidstafelperspectief. Hoe kun je met behulp van waarheidstafels controleren of een bepaalde gevolgtrekking logisch geldig is? Heel eenvoudig (maar

soms wel veel werk; in § 8.3 zullen we een andere methode presenteren): maak waarheidstafels voor alle premissen. Beschouw dan alleen die valuaties (rijen waarheidswaarden in de tafel) die overal een 1 hebben (dat wil zeggen: die alle premissen waar maken). Leveren al die valuaties ook voor de conclusie een 1 op, dan is de gevolgtrekking propositielogisch geldig, anders is ze dat niet.

We behandelen een voorbeeld. Is $((p \to q) \to (p \to r))$ een logisch gevolg van $(p \to (q \to r))$? Het antwoord is uit de volgende tafel af te lezen:

p	q	r	$p \to (q \to r)$		$(p \to q) \to (p \to r)$		
1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
1	0	1	1	1	0	1	1
0	0	1	1	1	1	1	1
1	1	0	0	0			
0	1	0	1	0	1	1	1
1	0	0	1	1	0	1	0
0	0	0	1	1	1	1	1

Je ziet dat zelfs voor zo'n simpel geval als dit het opstellen van de complete tafel een hele klus is. De conclusie uit de tafel:

$$(p \to (q \to r)) \models ((p \to q) \to (p \to r)).$$

Dit wil zeggen: de redenering van $(p \to (q \to r))$ naar $((p \to q) \to (p \to r))$ is propositielogisch geldig. Alleen de valuaties die er toe doen zijn in de tafel helemaal uitgewerkt.

Opgave 8.6 Laat met behulp van de waarheidstafelmethode zien dat het redeneerpatroon Modus Ponens (dat wil zeggen: concludeer uit de premissen φ en $(\varphi \to \psi)$ tot ψ) propositielogisch geldig is.

Opgave 8.7 Laat op dezelfde manier zien dat het patroon Modus Tollens (dat wil zeggen: concludeer uit de premissen $(\varphi \to \psi)$ en $\neg\psi$ tot $\neg\varphi$) propositielogisch geldig is.

Opgave 8.8 Idem voor de volgende redeneerpatronen:

1. Concludeer uit $(\varphi \to \psi)$ en $(\psi \to \chi)$ tot $(\varphi \to \chi)$.
2. Concludeer uit $(\varphi \vee \psi)$ en $(\varphi \to \chi)$ tot $(\chi \vee \psi)$.

Tot slot de waarheidstafelparafrase van *logische equivalentie* tussen formules: twee formules zijn logisch equivalent desda ze dezelfde waarheidstafel hebben.

Opgave 8.9 Laat zien met behulp van waarheidstafels dat de volgende formules per regel logisch equivalent zijn (weer: voor elke keuze van φ en ψ).

1. φ; $\neg\neg\varphi$
2. $\neg\varphi$; $\varphi \to (\psi \wedge \neg\psi)$
3. $\neg(\varphi \wedge \psi)$; $\neg\varphi \vee \neg\psi$
4. $\neg(\varphi \vee \psi)$; $\neg\varphi \wedge \neg\psi$.

Drie van deze logische equivalenties hebben speciale namen: 1. heet de wet van de dubbele negatie; 3. en 4. zijn de zogenaamde wetten van De Morgan.

Deze laatste opdracht was bedoeld om je een zeker 'gevoel' bij te brengen voor de logische equivalentie van formules uit de propositielogica. Maar zoals gezegd: de waarheidstafelmethode is vaak nogal omslachtig als test voor logische equivalentie en logische geldigheid. In de volgende paragraaf zullen we kennis gaan maken met een testmethode die eleganter is.

De volgende opgave is van belang bij het begrijpen van semantische tableaus:

Opgave 8.10 Wat $m \models \psi_1, \ldots, \psi_n$ betekent is duidelijk:

$$m \models \psi_1 \quad \text{en} \quad \ldots \quad \text{en} \quad m \models \psi_n.$$

Analoog definiëren we $m \not\models \psi_1, \ldots, \psi_n$ als:

$$m \not\models \psi_1 \quad \text{en} \quad \ldots \quad \text{en} \quad m \not\models \psi_n.$$

Laat zien:

- $m \models \psi_1, \ldots, \psi_n \iff m \models \psi_1 \wedge \cdots \wedge \psi_n$.
- $m \not\models \psi_1, \ldots, \psi_n \iff m \not\models \psi_1 \vee \cdots \vee \psi_n$.

8.2 Normaalvormen

Een normaalvorm is een standaardrepresentatie van een formule. In de propositielogica zijn er tenminste twee soorten normaalvormen, te weten *conjunctieve normaalvormen* en *disjunctieve normaalvormen*.

Normaalvormen zijn belangrijk. Zo liggen conjunctieve normaalvormen ten grondslag aan zg. clause sets. Clause sets vormen het standaard invoerformaat van veel stellingenbewijzers. Clause sets worden behandeld in Hoofdstuk 16 vanaf blz. 191.

Literals

Voordat normaalvormen kunnen worden besproken moet je weten wat een literal is.

Definitie 8.9 (Literal) *Een* literal *is een propositieletter, of de negatie van een propositieletter.*

De volgende formules zijn bijvoorbeeld literals: p, $\neg p$, P, $\neg P$, A, en $\neg A$. De volgende formules zijn bijvoorbeeld geen literals: $\neg\neg p$, $\neg\neg P$, $A \wedge P$, $\neg[A \wedge P]$, en $\neg[P \to \neg Q]$.

Literals zijn erg belangrijk in de logica. Behalve bij normaalvormen zijn ze essentieel in automatisch stellingenbewijzen (bv. in clause sets) en in logisch programmeren (bv. in Prolog).

Er is nog een gemeen trucje met literals, dat vaak niet wordt uitgelegd in logicaboeken, en waar sommigen moeite mee hebben. Dat is het volgende. Als we weten dat de formule φ een literal is, dan willen we vaak z'n *syntactisch complement* (kortweg: complement) kunnen

aanduiden, bijvoorbeeld als $\varphi = A$, dan is φ's complement $\neg A$, en als $\varphi = \neg A$, dan is φ's complement A. Het probleem is dat φ's complement *niet* kan worden aangeduid met $\neg\varphi$. (Ga na!) Een oplossing hiervoor is om φ's complement aan te duiden met $\sim\varphi$. Dus bijvoorbeeld $\sim Q = \neg Q$ en $\sim\neg Q = Q$. Heel belangrijk is nu om te beseffen dat "$\sim$" een hulpsymbool is uit de meta-taal, en in het bijzonder niet behoort tot de object-taal van de propositielogica. We hoeven dus bijvoorbeeld geen waarheidstafel op te stellen voor "$\sim$". Merk verder op dat de $\sim$-operator ook kan worden toegepast op formules die geen literal zijn, maar dat gebeurt eigenlijk nooit, behalve als oefening in de volgende opgave.

Opgave 8.11 Geef van de volgende formules aan of het literals zijn. Geef van alle formules het syntactisch complement, i.e., het resultaat van de $\sim$-operator.

1. P
2. $\neg P$
3. $\neg(\neg Q)$.
4. $\neg\neg Q$.
5. $\neg\neg\neg Q$.
6. $\neg^4 Q$ $(= \neg\neg\neg\neg Q)$.
7. $\neg^{17} Q$.
8. $\neg A \wedge P$.
9. $\neg^{42}(\neg A \wedge P)$.

Disjunctieve normaalvormen

Een *disjunctieve normaalvorm* (DNF) is een disjunctie van conjuncties van literals. Voorbeelden:

1. $(p \wedge \neg q) \vee (\neg p \wedge r \wedge s) \vee (\neg r \wedge \neg s)$
2. $(p \wedge \neg q) \vee (\neg p \wedge r \wedge s) \vee (\neg r \wedge r)$
3. $(p \wedge \neg q) \vee (\neg p \wedge r \wedge s) \vee \neg r$
4. $(p \wedge \neg q) \vee (\neg p \wedge r \wedge s)$
5. $p \vee \neg r$
6. $p \wedge \neg q$

DNF (1), (2) en (3) bestaan elk uit drie delen. DNF (4) en (5) bestaan uit twee, en DNF (6) bestaat uit één deel.

Je hebt waarschijnlijk opgemerkt dat (5) en (6) bijzondere DNFs zijn. Als we consequent haakjes zouden zetten, zoals bij (1-4), dan zou (5) geschreven worden als $(p) \vee (\neg r)$, terwijl (6) geschreven zou worden als $(p \wedge \neg q)$.

i) Een ander voorbeeld van een bijzondere DNF is de *lege* DNF, welke een DNF is met nul componenten.

ii) Omdat een DNF een disjunctie is, is deze geldig als en slechts als tenminste één component geldig is.

iii) Vanwege *i)* en *ii)* is de lege DNF altijd ongeldig.

iv) Omdat elk onderdeel van een DNF een conjunctie is, is een onderdeel van een DNF vervulbaar als en slechts als er geen complementair paar literals in voorkomt.

v) Vanwege *ii)* en *iv)* is een DNF vervulbaar als en slechts als het een component heeft zonder complementaire literals. DNFs kunnen dus zeer makkelijk worden gecontroleerd op vervulbaarheid.

Iedere propositie kan worden omgeschreven naar een logisch gelijkwaardige DNF. Daar zijn verschillende methoden voor. Sommige methoden zijn inzichtelijk maar computationeel inefficiënt. Andere methoden zijn efficiënt maar weer minder inzichtelijk. Waarschijnlijk de meest intuïtieve methode om DNFs te bereken is de methode die gebruik maakt van waarheidstabellen.

DNFs uit waarheidstabellen

Stel dat we

$$\varphi = (p \wedge q) \rightarrow \neg(q \vee \neg p) \tag{1}$$

in DNF willen schrijven. Met behulp van φ's waarheidstabel

(	p	$\wedge$	q	)	$\rightarrow$	$\neg$	(	q	$\vee$	$\neg$	p	)
	0	0	0		1	1		0	1	1	0	
	0	0	1		1	0		1	1	1	0	
	1	0	0		1	1		0	0	0	1	
	1	1	1		0	0		1	1	0	1	

zien we dat de eerste drie rijen φ waarmaakt, terwijl de laatste rij φ onwaar maakt. Omdat in een waarheidstabel alle mogelijke combinaties van p- en q-waarden voorkomen is

$$(\neg p \wedge \neg q) \vee (\neg p \wedge q) \vee (p \wedge \neg q)$$

een DNF die logisch gelijkwaardig is met φ. Misschien is dit niet de meest korte DNF, maar ons punt was hier om aan te tonen hoe een DNF systematisch gefabriceerd kan worden, niet om onmiddellijk de kortste te vinden.

Opgave 8.12 1. Is het mogelijk om een kortere DNF te vinden voor (1)?

2. Zijn DNFs uniek, i.e., bestaat er bij elke formule precies één normaalvorm?

DNFs door herschrijven

Een andere methode voor het berekenen van een DNF, is door het systematisch elimineren van implicaties, equivalenties, gevolgd door het wegwerken van alle haakjes met behulp van de distributieve wetten voor conjuncties en disjuncties. In meer detail:

1. Elimineer "$\rightarrow$" en "$\equiv$" door het toepassen van de herschrijf-regels $A \rightarrow B \rightsquigarrow \neg A \vee B$ en $A \equiv B \rightsquigarrow A \wedge B \vee \neg A \wedge \neg B$. (Voor de laatste zijn al geen haakjes nodig, dus dat is makkelijk.)

2. Elimineer haakjes door het herhaald toepassen van distributieve wetten, de wetten van De Morgan, en

door het weghalen van dubbele negaties:

$$\begin{aligned} A \wedge (B \vee C) &\rightarrow A \wedge B \vee A \wedge C \\ (A \vee B) \wedge C &\rightarrow A \wedge C \vee B \wedge C \\ \neg[A \wedge B] &\rightarrow \neg A \vee \neg B \\ \neg[A \vee B] &\rightarrow \neg A \wedge \neg B \\ \neg\neg A &\rightarrow A \end{aligned}$$

Uiteindelijk eindig je met een uitdrukking zoals bijvoorbeeld

$$a_1 \wedge \neg a_2 \wedge \neg a_3 \vee a_4 \wedge a_5 \vee \neg a_6 \wedge a_7 \vee \ldots.$$

Deze uitdrukking staat in DNF omdat "$\wedge$" prioriteit heeft over "$\vee$," wat misschien duidelijk wordt als we het als volgt opschrijven:

$$(a_1 \wedge \neg a_2 \wedge \neg a_3) \vee (a_4 \wedge a_5) \vee (\neg a_6 \wedge a_7) \vee \ldots.$$

Merk op dat deze procedure eventueel kan worden geautomatiseerd (wat in de informatica essentieel is).

Conjunctieve normaalvormen

A *conjunctieve normaalvorm* (CNF) is een conjunctie van disjuncties van literals.

i) Een bijzonder voorbeeld van een CNF is de *lege* CNF. Dat is een CNF met nul componenten.

ii) Omdat een CNF een conjunctie is, is een CNF waar a.e.s.a. alle onderdelen waar zijn.

iii) Vanwege *i)* en *ii)* is de lege CNF waar. (Als een CNF geen onderdelen heeft, zijn zonder meer alle onderdelen waar.)

iv) Omdat elk onderdeel van een CNF een disjunctie van literals is, is een onderdeel van een CNF waar a.e.s.a. er twee complementaire literals in voorkomen.

v) Vanwege *ii)* en *iv)* is een CNF tautologie (i.e., logische geldigheid) a.e.s.a in alle onderdelen een complementair paar literals voorkomt. CNFs kunnen dus zeer makkelijk worden gecontroleerd op logische geldigheid.

Om te zien dat elke propositie in CNF kan worden geschreven, nemen we een willekeurige propositie φ. Omdat we inmiddels weten dat elke propositie in DNF kan worden geschreven, kunnen we in het bijzonder $\neg\varphi$ in DNF schrijven:

$$\neg\varphi = (a_1 \wedge a_2 \wedge \ldots) \vee (b_1 \wedge b_2 \wedge \ldots) \vee \ldots$$

Met de wetten van De Morgan (blz. 88) volgt nu

$$\begin{aligned} \varphi &\equiv \neg[\neg\varphi] \\ &\equiv \neg[(a_1 \wedge a_2 \wedge \ldots) \vee (b_1 \wedge b_2 \wedge \ldots) \vee \ldots] \\ &\equiv \neg(a_1 \wedge a_2 \wedge \ldots) \wedge \neg(b_1 \wedge b_2 \wedge \ldots) \wedge \neg(\ldots) \wedge \ldots \\ &\equiv (\neg a_1 \vee \neg a_2 \vee \ldots) \wedge (\neg b_1 \vee \neg b_2 \vee \ldots) \wedge \ldots \qquad (2) \end{aligned}$$

Het verwijderen van dubbele negaties uit (2) levert φ in CNF op.

Opgave 8.13 1. Bepaal een DNF voor de volgende formules met behulp van een waarheidstabel.

(a) $p \rightarrow (q \rightarrow p)$

(b) $a \wedge (b \wedge (c \vee d))$

(c) $\neg(\neg(m))$

(d) $p \vee \neg p$

(e) $p \wedge \neg p$

2. Onderdeel (1), nu door middel van herschrijven.

3. Beschrijf een eenvoudig algoritme dat kan bepalen of een DNF vervulbaar is. Wat is grofweg de looptijd van dit algoritme, uitgedrukt in de lengte van de input?

4. Onderdeel (1), nu met CNFs.

5. Onderdeel (1), nu met CNFs en herschrijven.

6. Beschrijf een eenvoudig algoritme dat kan bepalen of een CNF logisch geldig is. Dezelfde vraag over looptijd als in Onderdeel (3).

7. Beschouw de formule

$$(a_1 \vee a_2) \wedge \cdots \wedge (z_1 \vee z_2). \qquad (3)$$

Deze formule heeft lengte $26c$, voor een of andere constante c.

(a) Bepaal c.

(b) Bepaal een kortste DNF voor (3). (Hint: probeer eerst een DNF voor $(a_1 \vee a_2) \wedge (b_1 \vee b_2)$ te construeren. Probeer dan wat je gedaan hebt te generaliseren.) Hoe weet je dat het de kortste DNF is?

(c) Wat is de lengte van de formule gevonden in (7b)? (Tel in literals.)

(d) Generaliseer je antwoord als je begint met een formule bestaande uit $2 \times n$ literals, in plaats van 2×26.

(e) Wat vertelt dit resultaat over de conversie van CNF naar DNF?

(f) Trek een algemene conclusie op basis van het vorige antwoord.

8. Beschouw de formule

$$(a \wedge \cdots \wedge z) \vee (\neg a \wedge \cdots \wedge \neg z) \qquad (4)$$

(a) Dezelfde vragen als bij (7).

(b) Probeer te verklaren waarom de conversie van (7) naar CNF zo sterk groeit.

Er zijn makkelijker manieren om normaalvormen te berekenen, maar daarvoor moet je bekend zijn met semantische tableaus.

8.3 Semantische tableaus

De semantische tableaumethode is in eerste instantie helemaal niet bedoeld voor het construeren van normaalvormen, maar is ontworpen voor het testen van redeneringen. Voor dit doel is de tableaumethode eleganter en meestal (maar niet altijd!) *economischer* dan het uitschrijven van waarheidstafels, omdat in het algemeen niet alle mogelijke valuaties hoeven te worden uitgewerkt. De tableau-methode heeft bovendien het voordeel dat zij kan worden uitgebreid tot een geldigheidstest voor predikatenlogische redeneringen (zie § 12). De waarheidstafelmethode voor het controleren van geldigheid is rechttoe, rechtaan: onderzoek gewoon alle mogelijkheden. Bij het maken van een waarheidstafel ter controle van een redenering

$$\varphi_1, \ldots, \varphi_n \models \psi$$

moeten we, als er m verschillende propositieletters voorkomen in de premissen en de conclusie, 2^m verschillende waarderingen (rijen in de waarheidstafel) gaan onderzoeken. Bij de tableaumethode kunnen we hier in veel gevallen op bezuinigen.

De semantische tableaumethode werkt niet direct maar *indirect*: wat er gebeurt is dat er systematisch wordt gezocht naar een *tegenvoorbeeld* tegen de redenering. De methode werkt als volgt: veronderstel dat de conclusie *onwaar* is, en dat alle premissen waar zijn. Wat moet er dan aan de hand zijn...? Aldus terugredenerend worden alle wegen die mogelijkerwijs leiden naar een tegenvoorbeeld uitgeplozen. Lukt het om zo'n tegenvoorbeeld te construeren (dat wil zeggen een valuatie te vinden die de premissen waar maakt en de conclusie onwaar), dan is de redenering kennelijk ongeldig. Lukt het niet dan mogen we, dankzij het feit dat het zoeken systematisch geschiedde, concluderen dat de redenering geldig is. Op dezelfde manier kunnen we tautologieën op het spoor komen, door het systematisch zoeken naar een valuatie die de kandidaat-tautologie *onwaar* maakt (een tautologie is immers een formule die logisch volgt uit de lege premissenverzameling).

We zullen de semantische tableaumethode introduceren aan de hand van voorbeelden. Om de voorbeelden te begrijpen moet je weten wat een semantisch tableau is: een plaatje dat is onderverdeeld in een linker- en een rechterkolom, met links en rechts verzamelingen formules. Links op elke regel staan de formules die waar moeten worden, rechts de formules die onwaar moeten worden (in het tegenvoorbeeld dat we proberen te construeren). Bij het testen van een redenering beginnen we met de premissen links te zetten en de conclusie rechts. Immers: in het tegenvoorbeeld waarnaar we op zoek zijn moeten de premissen waar worden en de conclusie onwaar. Hoe de verdere constructie van het tableau plaatsvindt moge blijken uit de voorbeelden die we nu gaan behandelen.

Voorbeeld 8.1 Is $p \wedge q \models p$?

Neem aan van niet:

waar	onwaar
$p \wedge q$	p

Het feit dat de formule $p \wedge q$ links in het tableau staat betekent dat die formule waar is, het feit dat p rechts staat betekent dat p onwaar is. Als $p \wedge q$ waar is, dan betekent dat dat zowel p als q waar moet zijn:

waar	onwaar
$p \wedge q$	p
p, q	

Dit kan echter niet: p staat nu zowel links als rechts in het tableau, en zou dus zowel waar als onwaar moeten zijn. Dit noteren we met een streep onderaan. We zeggen: het tableau *sluit*. De conclusie die we mogen trekken is dat het construeren van een tegenvoorbeeld tegen $p \wedge q \models p$ is mislukt. Met andere woorden: de redenering is geldig.

Semantische tableaus worden ook vaak als boomdiagrammen geschreven. Toepasselijk spreekt men dan ook wel van *semantische bomen*, niet te verwarren met constructiebomen of syntaxbomen. Het bovenstaande voorbeeld zou er dan zo uitzien:

$$\begin{array}{c} p \wedge q \ \circ \ p \\ \text{links-}\wedge \\ | \\ p, q \ \circ \\ \times \end{array}$$

Elke knoop in de boom bevat een cirkeltje. Links van het cirkeltje staan de formules die waar moeten worden gemaakt. Rechts van het cirkeltje staan de formules die onwaar moeten worden gemaakt. Formules die worden gedecomponeerd (of: ontleed) verschijnen een niveau lager in de boom. Als een tak sluit omdat er tegelijkertijd een formule waar en onwaar moeten worden gemaakt (hier q) wordt dat aangegeven met een kruisje $\times$.
Boomdiagrammen zijn overigens pas (veel) later in dit dictaat toegevoegd, daarom komen deze wat gefragmenteerd voor.

Voorbeeld 8.2 Is $p \vee q \models p$?

We beginnen het tableau weer als volgt:

waar	onwaar
$p \vee q$	p

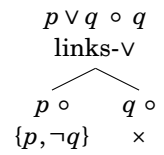
Dat $p \vee q$ waar is betekent dat minstens een van de twee volgende mogelijkheden moet zijn gerealiseerd: hetzij p is waar, hetzij q. Het feit dat er twee mogelijkheden zijn geven we in het tableau aan door middel van splitsing:

waar $p \vee q$		onwaar p	
waar	onwaar	waar	onwaar
p		q	

Een van de twee voortzettingen leidt tot een onmogelijke situatie. Die voortzetting leidt tot sluiting. Dit geven we weer aan door middel van een streep onderaan. De andere

voortzetting sluit niet. Die voortzetting levert een tegenvoorbeeld: q is waar, p is niet waar. Uit het bestaan van dit tegenvoorbeeld volgt dat de redenering ongeldig is.

In boomvorm:



Er zijn twee manieren om de disjunctie $p \vee q$ waar te maken, namelijk door p waar te maken (linkertak) of door q waar te maken (rechtertak). De linkertak is uitontwikkeld en levert een tegenmodel op, namelijk $\{p, \neg q\}$. De rechtertak sluit, omdat het onmogelijk is q zowel waar als onwaar te maken.

We moeten er aan denken dat we, om een tableau te controleren op sluiting, steeds het hele tableau van boven naar beneden moeten doorzoeken. Bij een opsplitsing van een tableau in tweeën loopt het oorspronkelijke tableau door in twee 'takken': het gedeelte voor de splitsing plus de ene voortzetting, en het gedeelte voor de splitsing plus de tweede voortzetting. Wanneer een van de twee tableauontwikkelingen zich wederom splitst hebben we in totaal drie takken, enzovoorts. Met een *subtableau* bedoelen we vanaf nu een volledige 'tak' in het tableau. Een subtableau *sluit* als er een formule φ zowel ergens onder 'waar' als ergens onder 'onwaar' voorkomt in dat subtableau. Het hele tableau sluit als elk subtableau sluit.

Voorbeeld 8.3 Is $p \models p \wedge q$?

Stel van niet:

waar	onwaar
p	$p \wedge q$

Als $p \wedge q$ onwaar is, dan moet minstens een van de twee volgende voorwaarden vervuld zijn: p onwaar, of q onwaar. Voor het tableau levert dit op:

waar p		onwaar $p \wedge q$	
waar	onwaar p	waar	onwaar q

Tegenvoorbeeld: p waar, q onwaar. De redenering is niet geldig.

Voorbeeld 8.4 Is $p \models p \vee q$?

Stel van niet:

waar	onwaar
p	$p \vee q$

Dat $p \vee q$ onwaar is, wil zeggen dat p en q beide onwaar moeten zijn:

waar	onwaar
p	$p \vee q$
	p, q

Het tableau sluit, de redenering is geldig.

Voorbeeld 8.5 Is $\neg\neg p \models p$?

Stel van niet:

waar	onwaar
$\neg\neg p$	p

De negatie van een bewering is waar wanneer die bewering zelf onwaar is, en de negatie van een bewering is onwaar wanneer de bewering zelf waar is. Dit geeft de volgende ontwikkeling van het tableau:

waar	onwaar
$\neg\neg p$	p
	$\neg p$
p	

Het tableau sluit; de redenering is geldig.

In de voorbeelden zijn tot nu toe de volgende regels voor de behandeling van operatoren geïntroduceerd:

– $\wedge$-links:

waar	onwaar
$\varphi \wedge \psi$	
φ, ψ	

– $\wedge$-rechts:

waar		onwaar $\varphi \wedge \psi$	
waar	onwaar φ	waar	onwaar ψ

– $\vee$-links:

waar $\varphi \vee \psi$		onwaar	
waar φ	onwaar	waar ψ	onwaar

– $\vee$-rechts:

waar	onwaar
	$\varphi \vee \psi$
	φ, ψ

– $\neg$-links:

waar	onwaar
$\neg\varphi$	
	φ

– $\neg$-rechts:

waar	onwaar
	$\neg\varphi$
φ	

– Het recept voor het afsluiten van een subtableau:

waar	onwaar
⋮	
φ	⋮
⋮	φ

of

waar	onwaar
	⋮
⋮	φ
φ	⋮

Dit wil zeggen: een *subtableau sluit* wanneer een formule φ zowel links als rechts in dat subtableau voorkomt. Let wel: een subtableau begint altijd helemaal bovenaan. Denk eraan dat φ geen atomaire formule hoeft te zijn. Een *tableau sluit* wanneer elk subtableau van dat tableau sluit.

We kunnen de bovenstaande regels gebruiken voor het construeren van tableaus voor redeneringen met $\wedge$, $\vee$ en $\neg$ als voegwoorden.

Voorbeeld 8.6 Is $\neg\varphi \vee \neg\psi \models \neg(\varphi \wedge \psi)$?

Stel van niet:

waar	onwaar
$\neg\varphi \vee \neg\psi$	$\neg(\varphi \wedge \psi)$

Het volledige tableau voor het testen van deze redenering ziet er als volgt uit:

waar $\neg\varphi \vee \neg\psi$, $\varphi \wedge \psi$, φ, ψ		onwaar $\neg(\varphi \wedge \psi)$	
waar $\neg\varphi$	onwaar φ	waar $\neg\psi$	onwaar ψ

Alle subtableaus van het tableau sluiten: de redenering is geldig. In boomvorm:

$\neg\varphi \vee \neg\psi \circ \neg(\varphi \wedge \psi)$
rechts-$\neg$
$\varphi \wedge \psi \circ$
links-$\wedge$
$\varphi, \psi \circ$
links-$\vee$ (op $\neg\varphi \vee \neg\psi$, immers φ en ψ zijn uitgeanalyseerd)

$\neg\varphi \circ$ links-$\neg$ $\circ\varphi$ ×
$\neg\psi \circ$ rechts-$\neg$ $\circ\psi$ ×

Merk op dat we bij het construeren van dit semantisch tableau zelf in de hand hebben in welke *volgorde* we bepaalde regels willen toepassen. We hebben eerst $\neg$-rechts toegepast, vervolgens $\wedge$-links, daarna $\vee$-links, en tenslotte in beide subtableaus $\neg$-links. We hadden ook kunnen beginnen met $\vee$-links:

$\neg\varphi \vee \neg\psi \circ \neg(\varphi \wedge \psi)$

$\neg\varphi \circ$ | $\circ\varphi$ | $\varphi \wedge \psi \circ$ | $\varphi, \psi \circ$ ×

$\neg\psi \circ$ | $\circ\psi$ | $\varphi \wedge \psi \circ$ | $\varphi, \psi \circ$ ×

Voor het eindresultaat maakt dat niet uit, maar in de praktijk is het handig om het toepassen van regels die tableau-splitsing tot gevolg hebben zo lang mogelijk uit te stellen. Zo wordt onnodig kopieerwerk voorkomen. (Zie laatste twee stappen in de laatste boom.)

Nog een voorbeeld. In een ongelukkig geval kan het voorkomen dat er voortdurend gesplitst moet worden:

$p, q, (p \to (q \to r)) \vee (p \to r) \circ r$
links-$\vee$

$p \to (q \to r) \circ$ links-$\to$: $q \to r \circ$ links-$\to$: $r \circ$ $\circ q$ ×; $\circ p$ ×
$p \to r \circ$ links-$\to$: $r \circ$ ×; $\circ p$ ×

Opgave 8.14 Laat met behulp van de semantische tableau methode zien dat de volgende formule-schema's regel voor regel logisch equivalent zijn:

1. $\varphi \wedge (\psi \vee \chi)$; $(\varphi \wedge \psi) \vee (\varphi \wedge \chi)$
2. $\varphi \vee (\psi \wedge \chi)$; $(\varphi \vee \psi) \wedge (\varphi \vee \chi)$.

Dit zijn de zogenaamde *wetten van de distributie*. Aanwijzing: om te laten zien dat twee formules φ en ψ logisch equivalent zijn moet je laten zien dat $\varphi \models \psi$ *en* dat $\psi \models \varphi$.

Opgave 8.15 Geef de regels voor a-links en a-rechts (a staat voor de exclusieve disjunctie).

Wanneer we nu nog regels voor de behandeling van $\to$ en $\leftrightarrow$ invoeren is onze tableaumethode compleet. Hoe $\to$-links behandeld moet worden volgt direct uit de waarheidstafel-definitie: $\varphi \to \psi$ is waar wanneer hetzij φ onwaar is, hetzij ψ waar. Dus:

– $\to$-links:

waar $\varphi \to \psi$		onwaar	
waar	onwaar φ	waar ψ	onwaar

De motivering voor $\to$-rechts moet je nu zelf kunnen bedenken. Hier is de regel:

– $\to$-rechts:

waar	onwaar
	$\varphi \to \psi$
φ	ψ

Opgave 8.16 Ga met behulp van een tableau waarin de regel α-rechts wordt gebruikt die je in de opdracht 8.15 heeft geformuleerd na of een redenering volgens het volgende schema geldig is:

$$\varphi \vee \psi \models \varphi \; a \; \psi.$$

Opgave 8.17 Geef de regels voor $\leftrightarrow$-links en $\leftrightarrow$-rechts.

Opgave 8.18 Laat met behulp van de semantische tableaumethode zien dat de volgende formule-schema's regel voor regel logisch equivalent zijn:

1. $\varphi \to (\psi \to \chi)$; $(\varphi \wedge \psi) \to \chi$
2. $\varphi \leftrightarrow \psi$; $(\varphi \to \psi) \wedge (\psi \to \varphi)$
3. $\varphi \to \psi$; $\neg\psi \to \neg\varphi$
4. $\varphi \to \neg\psi$; $\psi \to \neg\varphi$.

3. en 4. zijn de zogenaamde *wetten van de contrapositie*.

Dat de principes uit opdracht 8.18 in het redeneren een rol spelen blijkt bij voorbeeld uit het feit dat bij het *bewijzen* van een stelling van de vorm "Zus is het geval desda zo het geval is" er vaak wordt *uitgesplitst*. Er worden dan twee onderdelen bewezen:

– Ten eerste: als zus het geval is, dan is zo het geval.
– Ten tweede: als zo het geval is, dan is zus het geval.

Hier ligt natuurlijk de equivalentie uit onderdeel 2. van 8.18 aan ten grondslag.

Een principe dat bij het bewijzen van stellingen van de vorm "Als zus het geval is, dan is zo het geval" vaak wordt gebruikt is principe 3. uit 8.18. We kunnen dan contrapositie toepassen, en in plaats van de oorspronkelijke stelling bewijzen we: "Als zo *niet* het geval is, dan is zus *niet* het geval".

Opgave 8.19 Laat met behulp van de semantische tableaumethode zien dat de volgende formules tautologieën zijn:

1. $(p \wedge (p \to q)) \to q$
2. $(\neg q \wedge (p \to q)) \to \neg p$
3. $((p \to q) \wedge (q \to r)) \to (p \to r)$
4. $p \to (p \to p)$
5. $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$.

Aanwijzing: om met de tableaumethode te laten zien dat een formule φ een tautologie is dienen we te controleren of de aanname dat φ onwaar is tot een tegenspraak leidt. Dit houdt in dat we het tableau moeten beginnen door φ rechts in het tableau te zetten.

8.4 Correctheid

Hier aangekomen heb je hopelijk een groot aantal bewijzen geconstrueerd, en goede intuïtie gekregen voor de werking van semantisch tableaus.

Het construeren van vele tableaus leidt op enig moment tot de vraag of de tableaumethode correct is. Is de tableaumethode bijvoorbeeld krachtig genoeg om er alle geldige stellingen mee te kunnen afleiden? En als dat zo mocht zijn, is de tableaumethode dan niet per ongeluk té krachtig? I.e., kunnen we uitsluiten dat er met de tableaumethode per ongeluk onware stellingen worden afgeleid?

Deze vragen hebben betrekking op twee wenselijke eigenschappen. Laat Γ een eindige verzameling van formules, en φ een formule uit de propositielogica zijn.

– *Gezondheid.* Als $\Gamma \not\models \varphi$ dan blijft elk tableau voor $\Gamma \circ \varphi$ open. (Welke regels we ook toepassen en wat we ook proberen we krijgen het tableau niet "dicht".)
– *Volledigheid.* Als $\Gamma \models \varphi$ dan kunnen we elk tableau voor $\Gamma \circ \varphi$ sluiten. (Hoe dat moet is een ander verhaal.)

Als een afleidingssysteem *gezond* is, dan wil dat zeggen dat er geen onware stellingen kunnen worden afgeleid. Als een afleidingssysteem *volledig* is, dan wil dat zeggen dat het krachtig genoeg is om alle ware stellingen af te kunnen leiden. Een afleidingssysteem heet *correct* als en alleen als het gezond en volledig is. In een correct afleidingssysteem kunnen alle ware stellingen worden afgeleid, en is het onmogelijk om onware stellingen af te leiden.

Hoe bewijzen we nu dat de tableaumethode correct is? Om daar gevoel voor te krijgen is het allereerst belangrijk door te krijgen dat gezondheid in het algemeen makkelijk te bewijzen is.

Gezondheid

Om te laten zien dat de tableaumethode gezond is moet je laten zien dat, als je begint met $\Gamma \circ \varphi$ zó dat $\Gamma \not\models \varphi$, er bij uitbreiding van het tableau altijd tenminste één tak openblijft. Zie ook Fig. 8.1.

We voeren wat terminologie in. Een *knoop* is een punt in een tak

$$\varphi_1, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$

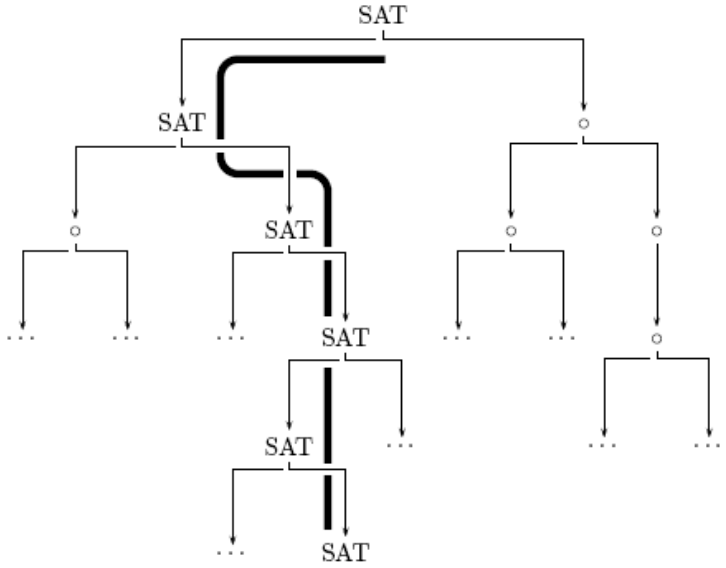
waarbij $\varphi_1, \ldots, \varphi_m$ alle formules zijn die op dat moment waar moeten worden gemaakt, en $\psi_1, \ldots, \psi_n$ alle formules zijn die op dat moment onwaar moeten worden gemaakt. Laten we een knoop *open* noemen als er een model m is, zó dat

$$m \models \varphi_1, \ldots, \varphi_k \;\text{ maar }\; m \not\models \psi_1, \ldots, \psi_n,$$

waarbij we $m \not\models \psi_1, \ldots, \psi_n$ moeten lezen als

$$m \not\models \psi_1 \quad \text{en} \quad \ldots \quad \text{en} \quad m \not\models \psi_n.$$

(Zie Opgave 8.10, blz. 88.) Een *open tak* is dan een tak met open knopen.



Figuur 8.1: Voortplanting van een model m door tenminste één tak van een semantisch tableau.

Stel $\Gamma \not\models \varphi$. Als we dit tot beginknoop van een nog te ontwikkelen tableau maken, dan heeft dit tableau, bestaande uit één knoop, per definitie een open knoop.

Laten we nu per inductie aannemen dat het tableau verder ontwikkeld is, en nog tenminste één open tak bezit, met als laatste open knoop

$$\varphi_1, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n.$$

Als die tak niet verder kan worden ontwikkeld zijn we klaar: de tak is open, kan niet verder worden ontwikkeld, en blijft dus open, en dat is wat we moesten bewijzen. Als die tak nog wél verder kan worden ontwikkeld, dan kan dat niet anders gebeuren dan met één van de tableauregels voor $\wedge$, $\vee$, $\neg$, of $\rightarrow$.

We bekijken alleen de gevallen links-$\wedge$ en rechts-$\wedge$. De rest verloopt analoog en laten we ter oefening.

Stel, de links-$\wedge$ regel werd toegepast. Dan ziet die open knoop er z.b.d.a.[1] uit als

$$\varphi_1^1 \wedge \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$

met een model, zeg m, zó dat

$$m \models \varphi_1^1 \wedge \varphi_1^2, \varphi_2, \ldots, \varphi_k \text{ maar } m \not\models \psi_1, \ldots, m \not\models \psi_n.$$

De stap zelf ziet er als volgt uit:

$$\begin{array}{c} \varphi_1^1 \wedge \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n \\ \text{links-}\wedge \\ | \\ \varphi_1^1, \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n \\ \vdots \end{array}$$

Makkelijk is na te gaan dat de volgende knoop nog steeds open is. Daarvoor moeten we het volgende controleren:

$$m \models \varphi_1^1, \varphi_1^2, \varphi_2, \ldots, \varphi_k \text{ maar } m \not\models \psi_1, \ldots, m \not\models \psi_n.$$

Dat is waar, immers als $m \models \varphi_1^1 \wedge \varphi_1^2$ dan uiteraard $m \models \varphi_1^1$ en $m \models \varphi_1^2$. De omringende formules $\varphi_2, \ldots, \varphi_k, \psi_1, \ldots, \psi_n$ doen daar niets aan af. (Ga na!) Het geval waarin links-$\wedge$ regel werd toegepast is nu afgehandeld.

Stel, de rechts-$\wedge$ regel werd toegepast. Dan ziet die open knoop er z.b.d.a. uit als

$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1 \wedge \psi_1^2, \psi_2, \ldots, \psi_n$$

met een model, zeg m, zó dat

$$m \models \varphi_1, \ldots, \varphi_k \text{ maar } m \not\models \psi_1^1 \wedge \psi_1^2, m \not\models \psi_2, \ldots, m \not\models \psi_n.$$

De eigenlijke stap ziet er uit als:

$$\begin{array}{c} \varphi_1, \ldots, \varphi_k \circ \psi_1^1 \wedge \psi_1^2, \psi_2 \ldots, \psi_n \\ \text{rechts-}\wedge \\ \varphi_1, \cdots \circ \psi_1^1, \psi_2, \ldots \qquad \varphi_1, \cdots \circ \psi_1^2, \psi_2, \ldots \\ \vdots \qquad\qquad\qquad \vdots \end{array}$$

Makkelijk is na te gaan dat tenminste één van de volgende knopen nog steeds open is:

$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1, \psi_2, \ldots, \psi_n$$
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^2, \psi_2, \ldots, \psi_n$$

Immers, als $m \not\models \psi_1^1 \wedge \psi_1^2$ dan uiteraard $m \not\models \psi_1^1$ of $m \not\models \psi_1^2$, de omringende formules $\varphi_1, \ldots, \varphi_k, \psi_2, \ldots, \psi_n$ doen daar niets aan af. (Ga na!)

We hebben nu alleen de gevallen links-$\wedge$ en rechts-$\wedge$ bekeken. Zoals gezegd verloopt de rest analoog. Dit laten we ter oefening. (Opgave 8.20, “naar beneden”.)

[1] “Zonder beperking der algemeenheid’, jargon voor: het kan ook andersom, maar dan verloopt de redenering net zo.

Volledigheid

Dit is andere koek. We moeten dan namelijk laten zien dat voor elke geldige semantische gevolgtrekking $\Gamma \models \varphi$ **elk** mogelijk tableau sluit. Nu is dit moeilijk te bewijzen. Immers, je moet dan van **alle** mogelijke takken laten zien dat ze gesloten kunnen worden. Dat is erg veel werk. We kunnen beter de contrapositie bewijzen. In dat geval moet je laten zien dat, als je begint met $\Gamma \circ \varphi$, en één tak wil niet sluiten, dat dan $\Gamma \not\models \varphi$.

Stel dus dat we een tableau aan de hand hebben waarvan één tak niet wil sluiten. Laten we veronderstellen dat deze tak helemaal uitontwikkeld is, dat wil zeggen dat elke mogelijk regel er op is toegepast. Dan bestaat het blad (eindpunt) van die tak uit een knoop met alleen maar atomen:

$$p_1, \ldots, p_k \circ q_1, \ldots, q_n$$

Dat levert een concreet model m op met

$$m(p_1) = \cdots = m(p_k) = 1 \text{ en } m(q_1) = \cdots = m(q_n) = 0.$$

Er kan worden gecontroleerd dat m er voor zorgt dat elke knoop boven dit blad ook open is (Opgave 8.20, "naar boven"). Dus m plant zich voort, terug door het tableau heen naar boven, naar de top-knoop (zie Fig. 8.1). We wisten al dat de bovenliggende tak open is, maar nu weten we meer, namelijk dat het model in een open tak per knoop niet steeds hoeft te wisselen, maar dat één enkel model m al ervoor kan zorgen dat een hele open tak open blijft. In het bijzonder geldt voor de top-knoop dat $m \models \Gamma$ maar $m \not\models \varphi$. Ergo, $\Gamma \not\models \varphi$. □

Opgave 8.20 Bewijs de volgende implicaties.

1. **links-∧, naar beneden** Als
$$\varphi_1^1 \wedge \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
open is, dan is het kind
$$\varphi_1^1, \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
ook open.
2. **links-∧, naar boven** Omgekeerd: als het kind
$$\varphi_1^1, \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
open is, dan is de ouder
$$\varphi_1^1 \wedge \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
ook open.
3. **rechts-∧, naar beneden** Als
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1 \wedge \psi_1^2, \psi_2 \ldots, \psi_n$$
open is, dan is één van de kinderen
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1, \psi_2, \ldots, \psi_n$$
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^2, \psi_2, \ldots, \psi_n$$
ook open.
4. **rechts-∧, naar boven** Omgekeerd: als één van de kinderen
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1, \psi_2, \ldots, \psi_n$$
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^2, \psi_2, \ldots, \psi_n$$
open is, dan is de ouder
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1 \wedge \psi_1^2, \psi_2 \ldots, \psi_n$$
ook open.
5. **rechts-∨, naar beneden** Als
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1 \vee \psi_1^2, \psi_2, \ldots, \psi_n$$
open is, dan is het kind
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1, \psi_1^2, \psi_2, \ldots, \psi_n$$
ook open.
6. **rechts-∨, naar boven** Omgekeerd: als het kind
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1, \psi_1^2, \psi_2, \ldots, \psi_n$$
open is, dan is de ouder
$$\varphi_1, \ldots, \varphi_k \circ \psi_1^1 \vee \psi_1^2, \psi_2, \ldots, \psi_n$$
ook open.
7. **links-∨, naar beneden** Als
$$\varphi_1^1 \vee \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1 \ldots, \psi_n$$
open is, dan is één van de kinderen
$$\varphi_1^1, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
$$\varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
ook open.
8. **links-∨, naar boven** Omgekeerd: als één van de kinderen
$$\varphi_1^1, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
$$\varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$
open is, dan is de ouder
$$\varphi_1^1 \vee \varphi_1^2, \varphi_2, \ldots, \varphi_k \circ \psi_1 \ldots, \psi_n$$
ook open.
9. Bewijs de voortplantingsregel naar beneden voor negatie.
10. Bewijs de voortplantingsregel naar boven voor negatie.
11. Bewijs de voortplantingsregels voor implicatie.

Normaalvormen met tableaus

Normaalvormen werden geïntroduceerd in Sectie 8.2 blz. 88. Het construeren van een DNF met waarheidstafels is echter bewerkelijk.

Het is relatief onbekend dat tableaus ook uitermate geschikt zijn om vooral handmatig (met pen en papier) normaalvormen te construeren. Het is meestal ook minder werk.[2]

Voorbeeld 8.7 (een DNF uit een tableau) Stel dat we een DNF willen berekenen van $\varphi = (p \to q) \wedge (r \wedge s)$. Dit kan worden gedaan door φ links van de scheider "$\circ$" te plaatsen als begin van een te ontwikkelen tableau. We proberen φ dus waar te maken. Om te zien welke atomen φ waar maken, ontwikkelen we het tableau helemaal uit:

$$\begin{array}{c} (p \to q) \wedge (r \wedge s) \circ \\ \text{links-}\wedge \\ | \\ (p \to q), (r \wedge s) \circ \\ \text{links-}\wedge \\ | \\ (p \to q), r, s \circ \\ \text{links-}\to \\ \diagup \quad \diagdown \\ r, s \circ p \qquad q, r, s \circ \end{array}$$

De formule φ is nu waar a.e.s.a. een van de bladeren waar is. Dus φ is waar a.e.s.a. $r \wedge s \wedge \neg p$ waar is, of $q \wedge r \wedge s$ waar is. Dus

$$\varphi \equiv (r \wedge s \wedge \neg p) \vee (q \wedge r \wedge s).$$

Besef wel dat deze methode eist dat alle takken volledig worden uitontwikkeld, iets dat niet altijd nodig is als tableaus worden gebruikt om de logische geldigheid van formules te verifiëren. □

In Voorbeeld 8.7 werd getoond hoe DNFs kunnen worden berekend met semantische tableaus. CNFs kunnen ook worden berekend met semantische tableaus.

Voorbeeld 8.8 (een CNF uit een tableau) Stel dat we een CNF van $\varphi = (p \to q) \wedge (r \wedge s)$ willen berekenen. Dit kan door φ aan de rechterkant van de scheider "$\circ$" van een beginnend tableau te zetten. We proberen φ dus onwaar te maken.

$$\begin{array}{c} \circ (p \to q) \wedge (r \wedge s) \\ \text{rechts-}\wedge \\ \diagup \quad \diagdown \\ \circ p \to q \qquad \circ r \wedge s \\ \text{rechts-}\to \qquad \text{rechts-}\wedge \\ | \qquad\qquad \diagup \quad \diagdown \\ p \circ q \qquad\quad \circ r \quad \circ s \end{array}$$

Alle bladeren die tegenvoorbeelden representeren moeten worden ontkent om te voorkomen dat φ onwaar wordt. Dus

$$(\neg p \vee q) \wedge r \wedge s$$

is een CNF voor φ. Merk op dat er andere logisch equivalente, en zelfs kortere, CNFs kunnen bestaan. We merken nogmaals op dat deze methode alleen werkt als alle takken word gecompleteerd. □

Opgave 8.21 1. Opgave 8.13, blz. 90, Onderdeel (1), nu met tableaus.

2. Idem, nu met CNFs en tableaus.

Opgave 8.22 1. Bepaal DNFs voor de formules uit Opgave 8.3, blz. 87.

2. Idem, nu met CNFs.

3. Bepaal DNFs voor de formules uit Opgave 8.2.

4. Idem, nu met CNFs.

Opgave 8.23 Zijn DNFs (of CNFs) uniek? Zo ja, geef een bewijs. Zo nee, geef een tegenvoorbeeld, en onderzoek of er voorwaarden kunnen worden opgesteld waaronder DNFs (of CNFs) wel uniek zijn. (Hint: herlees nog eens de sectie "DNFs uit waarheidstabellen".)

Zijn er nog makkelijker manieren om normaalvormen te berekenen? Nee, niet echt.

Stel dat de conversie naar, bijvoorbeeld, CNF wel makkelijk ging. Dan zouden we proposities eenvoudig op logische geldigheid kunnen controleren door ze eerst naar CNF te converteren (makkelijk) om vervolgens de verkregen CNF te controleren op logische geldigheid (makkelijk, zie boven). Echter, het is in de informatica algemeen bekend dat het geldigheidsprobleem in de propositielogica een NP-volledig probleem is. Er zijn zeer, zeer veel NP-volledige problemen, die allemaal makkelijk onderling naar elkaar te vertalen (dus even moeilijk) zijn, en waarvan men tot op de dag van vandaag een zeer sterk vermoeden heeft dat deze problemen niet makkelijk (officieel: in polynomiale tijd) zijn op te lossen.

Een dergelijk soort verhaal geldt voor DNFs. Stel dat de conversie naar DNF makkelijk ging. Dan zouden we proposities eenvoudig op vervulbaarheid kunnen controleren door ze eerst naar DNF te converteren (makkelijk) om vervolgens de verkregen DNF te controleren op vervulbaarheid (makkelijk, zie boven). Het is in de informatica algemeen bekend dat het vervulbaarheidsprobleem in de propositielogica een co-NP-volledig probleem is. Zie verder de literatuur.

8.5 Functionele volledigheid

Een belangrijke vraag die over connectieven van de propositielogica kan worden gesteld is de vraag welke connectieven in termen van welke andere kunnen worden gedefinieerd. Een voorbeeld: het invoeren van een speciaal teken a voor het exclusieve of is niet nodig, omdat $\varphi \; a \; \psi$ kan worden weergegeven als $\neg(\varphi \leftrightarrow \psi)$. In § 9.6 hebben we gezien hoe $\wedge$, $\vee$ en $\leftrightarrow$ kunnen worden uitgedrukt in termen van $\to$ en $\neg$ alleen.

[2] Er zijn gevallen te bedenken waarvoor het meer werk is.

Op deze manier kun je nog meer connectieven wegwerken: $\leftrightarrow$ is definieerbaar in termen van $\rightarrow$ en $\wedge$; $\rightarrow$ is definieerbaar met behulp van $\neg$ en $\vee$. Een verzameling connectieven in termen waarvan je alle waarheidstafelpatronen kunt uitdrukken heet *functioneel volledig*. Iets formeler:

Definitie 8.10 *Een verzameling connectieven C heet* **functioneel volledig** *desda er voor elk natuurlijk getal* $n > 0$ *en voor elke functie*

$$f : \{0,1\}^n \rightarrow \{0,1\}$$

een formule φ *is met n propositieletters die alleen de connectieven uit C bevat en die het waarheidstafelpatroon van* f *heeft.*

Stelling 8.1 *De verzameling* $\{\wedge, \vee, \neg\}$ *is functioneel volledig.*

Bewijs: We leveren het bewijs aan de hand van een voorbeeld. Laat f een functie van $\{0,1\}^2$ naar $\{0,1\}$ zijn met bij voorbeeld als functievoorschrift:

$$\begin{aligned} f(1,1) &= 1 \\ f(0,1) &= 1 \\ f(1,0) &= 0 \\ f(0,0) &= 0. \end{aligned}$$

We construeren nu een formule φ die twee propositieletters p en q bevat en als waarheidstafel het patroon van f heeft, dat wil zeggen:

	p	q	φ
I_1	1	1	1
I_2	0	1	1
I_3	1	0	0
I_4	0	0	0

De formule φ mag volgens de spelregels alleen de voegwoorden $\wedge$, $\vee$ en $\neg$ bevatten. Merk op dat een argumentenpaar van f overeenkomt met een *interpretatie* van p en q. Zo'n interpretatie kan worden omschreven met behulp van een simpele conjunctie: voor de eerste interpretatie, I_1, is de omschrijving $p \wedge q$; I_2 kan worden omschreven als $\neg p \wedge q$; I_3 als $p \wedge \neg q$; I_4 als $\neg p \wedge \neg q$. Alleen I_1 en I_2 maken φ waar, dus we kunnen voor φ de disjunctie nemen van de corresponderende conjuncties: $(p \wedge q) \vee (\neg p \wedge q)$. Ga zelf na dat deze formule het gewenste waarheidstafelpatroon heeft.

Het is nu duidelijk hoe we voor elke gegeven tweeplaatsige waarheidsfunctie f een geschikte φ vinden: neem de disjunctie van de formules die horen bij de interpretaties die 1 moeten opleveren. In één geval werkt dit procédé niet: het geval dat f louter nullen als waarden heeft. In dat geval voldoet de formule $(p \wedge \neg p) \vee (q \wedge \neg q)$ (ga dit na).

Het niet moeilijk om in te zien hoe de geschetste methode kan worden toegepast voor willekeurige waarden van n. Als $n = 1$ dan hebben we één propositieletter p nodig, en kunnen we de interpretaties beschrijven met de formules p en $\neg p$. Als $n = 3$ dan zijn er drie propositieletters p, q en r nodig, en acht conjuncties om de acht mogelijke interpretaties te beschrijven: $p \wedge q \wedge r$, $\neg p \wedge q \wedge r$, $p \wedge \neg q \wedge r$, enzovoort. In het algemeen: gebruik n propositieletters en 2^n formules die elk een conjunctie zijn van (negaties van) elk van die n propositieletters, en neem voor φ de disjunctie over alle conjuncten die waar opleveren; als zulke conjuncten er niet zijn, neem dan de disjunctie van $p \wedge \neg p$, voor alle propositieletters p. □

Opgave 8.24 Laat met behulp van stelling 8.1 zien dat $\{\wedge, \neg\}$, $\{\rightarrow, \neg\}$, $\{\vee, \neg\}$ functioneel volledig zijn.

Opgave 8.25 Laat zien dat $\{\wedge, \vee\}$ niet functioneel volledig is.

Aanwijzing: je hoeft alleen te laten zien dat negatie niet uit te drukken valt.

Opgave 8.26 Laat zien dat $\{\neg, \leftrightarrow\}$ niet functioneel volledig is.

Opgave 8.27 (*) Laat zien dat $\{\wedge, \rightarrow\}$ niet functioneel volledig is. (Dit is moeilijker dan de vorige opgave en het volgt er ook niet uit, in weerwil van een oppervlakkige gelijkenis.)

Aanwijzing 1: met Stelling 8.1 volstaat het te zoeken naar een logische formule opgebouwd uit $\wedge$, $\vee$ en $\neg$, die niet kan worden uitgedrukt met $\{\wedge, \rightarrow\}$.

Aanwijzing 2: probeer eerst te ontdekken welke eigenschappen $\wedge$ en $\rightarrow$ bezitten, om dan vervolgens een formule te vinden die deze eigenschap mist.

Aanwijzing 3: kijk wat de connectieven $\wedge$ en $\rightarrow$ doen als beide deel-formules (van $\wedge$ of $\rightarrow$) waar zijn. Vergelijk dit met het gedrag van het negatie-connectief.

Interessant is dat zelfs één enkel connectief functioneel volledig kan zijn. De volgende connectieven kunnen het allebei (ze heten respectievelijk—naar hun uitvinders—de Quine dolk '†' en de Sheffer streep '|'):

†	1	0
1	0	0
0	0	1

|	1	0
1	0	1
0	1	1

Zoals je uit deze waarheidstafels kunt aflezen betekent $\varphi \dagger \psi$ 'noch φ noch ψ', terwijl $\varphi | \psi$ staat voor 'niet zowel φ als ψ'. Een voorbeeld van omschrijving: $\neg\varphi$ wordt met behulp van de Quine dolk $\varphi \dagger \varphi$, en met behulp van de Sheffer streep: $\varphi | \varphi$.

Opgave 8.28 Omschrijf de connectieven $\wedge$, $\vee$, $\rightarrow$ en $\leftrightarrow$ met behulp van alleen † en met behulp van alleen | .

Opgave 8.29 (*) Laat zien dat † en | de *enige* volledige binaire operatoren zijn.

Opgave 8.30 Geef semantische tableauregels voor de Quine dolk † en de Sheffer streep | .

Hoofdstuk 9

Bewijssystemen voor de propositielogica

De logica wordt traditioneel vooral gezien als gevolgtrekkingsleer, de discipline die zich bezighoudt met het construeren, beoordelen en systematiseren van (soms gecompliceerde) redeneringen. We hebben in het voorgaande hoofdstuk kennisgemaakt met de taal en de semantiek van de propositielogica en met het semantische geldigheidsbegrip ($\Gamma \models \varphi$), maar met het werkelijke redeneren heeft dit nog niet direct te maken; dat φ door Γ semantisch wordt geïmpliceerd wil niet zeggen dat we over een bewijs van φ uit Γ beschikken.

Wat is een bewijs eigenlijk? Een overtuigende redenering; daarbij denken we meestal aan een keten van redeneerstappen die ieder op zich een grote mate van vanzelfsprekendheid bezitten. Maar "vanzelfsprekend" is dan nog een vaag en discutabel begrip. In de formele behandeling van de logica gaat men er van uit dat de elementaire redeneerstappen van een afleiding instanties zijn van een (gering) aantal vaste basisredeneerpatronen. Er zijn verschillende mogelijkheden voor de keuze van deze patronen; verschillende keuzes leiden tot verschillende logische systemen.

In dit hoofdstuk worden twee bewijssystemen behandeld. Eerst behandelen we Frederic Fitch's systeem van natuurlijke deductie, daarna behandelen we David Hilbert's systeem. Beide systemen verschillen nogal van elkaar, en hebben elk zo hun eigen voor- en nadelen. Zo is het bijvoorbeeld nog doenbaar om in Fitch' systeem afleidingen te construeren (sommige zeggen "te vinden") voor geldige gevolgtrekkingen ($\Gamma \models \varphi$). Bovendien zien deze afleidingen er intuïtief uit, vandaar de naam "natuurlijke deductie". Een nadeel van natuurlijke deductie is dat het veel regels (ongeveer twee handen vol, afhankelijk hoe er geteld wordt) en een subbewijs-mechanisme bevat. Daardoor is het relatief iets moeilijker om over natuurlijke deductie te redeneren en er bijvoorbeeld eigenschappen over te bewijzen.

Hilbert's minimalistische systeem is met drie axiomaschema's en één afleidingsregel door zijn eenvoud beter geschikt om er *over* te redeneren (meta-theorie), maar veel minder geschikt om er *mee* te redeneren, i.e., om er afleidingen mee te genereren (object-theorie). Daarom beginnen we toch maar eerst met natuurlijke deductie.

9.1 Natuurlijke deductie

Fitch' systeem [Fitch 1952] wordt natuurlijke deductie genoemd, omdat het, zoals eerder aangegeven, de pretentie heeft nauw aan te sluiten bij het natuurlijk redeneren. Systemen van natuurlijke deductie zijn omstreeks 1930 ingevoerd door de Duitse logicus Gerhard Gentzen en de Poolse logicus Stanisław Jaśkowski. Door Gentzen werd ook nog een andere benadering van het afleidbaarheidsbegrip voorgesteld, de zogenaamd "sequenten-calculi". Deze, zeer aan natuurlijke deductie verwante, aanpak staat recent nogal in de belangstelling in de theoretische informatica.

Van natuurlijke deductie zijn verschillende varianten in omloop. De hier gebruikte variant is ontleend aan het boek "Taal, Logica en Betekenis, Deel 1" van L.T.F. Gamut [Gamut 1982].[1]

In het systeem van natuurlijke deductie corresponderen de elementaire redeneerstappen met de betekenis van de verschillende logische symbolen, waarbij ieder symbool in principe aanleiding geeft tot het formuleren van twee regels:

1. Een introductieregel (I) die aangeeft wanneer tot een formule met het bewuste connectief als hoofdteken kan worden geconcludeerd.

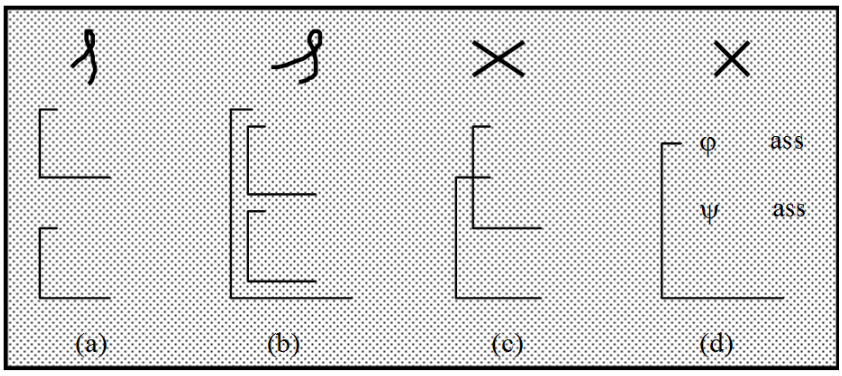
2. Een gebruiksregel (G) die aangeeft hoe een formule met dit connectief als hoofdteken kan worden gebruikt; bij toepassing van de bijbehorende gebruiksregel zal het connectief verdwijnen, vandaar dat gebruiksregels ook wel eliminatieregels worden genoemd.

Een bewijs of afleiding in natuurlijke deductie zal er nu als volgt gaan uitzien. Het is in beginsel een eindige lijst onder elkaar staande genummerde formules.

1.	φ_1
2.	φ_2
..	...
..	...
n.	φ_n

Achter elke formule φ_i staat een verklaring, waarmee wordt aangegeven op grond waarvan tot die formule geconcludeerd is. Daarvoor zijn twee mogelijkheden: óf φ_i

[1] Gamut is een pseudoniem voor een collectief van logici uit **G**roningen, **Am**sterdam, en **Ut**recht. "Taal, Logica en Betekenis" is alleen nog maar verkrijgbaar in de Engelse uitgave: Logic, Language and Meaning: Introduction to logic, The University of Chicago Press, 1991.) Je vindt er een iets uitgebreidere behandeling dan hier gegeven wordt.



Figuur 9.1: Intrekken van aannamen in een natuurlijke deductie.

is een aanname, in dat geval staat er "ass" achter φ_i (voor "assumptie"), óf φ_i is verkregen door toepassing van één of andere regel op formules boven φ_i, die dus al afgeleid waren. In het laatste geval worden achter φ_i de afleidingsregel en de (nummers van de) gebruikte formules aangegeven.

De eerste formule van een bewijs kan altijd alleen maar een aanname zijn. De laatste formule, φ_n, heet de conclusie van het bewijs.

Er zijn twee afleidingsregels die het geschetste beeld nog iets zullen compliceren, omdat bij hun toepassing een aanname wordt ingetrokken. Een ingetrokken aanname heeft een beperkte geldigheid: er kan slechts naar gerefereerd worden vanaf het moment dat de aanname gemaakt wordt, tot deze wordt ingetrokken. Dit geldigheidsgebied wordt grafisch aangegeven met behulp van een extra kantlijn.

1.	φ_1	ass
..	...	...
..	...	...
i.	φ_i	ass
..	...	...
..	...	...
j.	φ_j	...
$j+1$.	...	...
..	...	...
n.	φ_n	...

In deze afleiding mag naar de aanname φ_i dus slechts gerefereerd worden in de stappen $i+1$ tot en met j; vanaf stap $j+1$ niet meer, dan is φ_i niet meer geldig. Deze beperking geldt niet alleen φ_i, maar ook de formules die van φ_i afhangen: de stappen i tot en met j vormen een soort sub-afleiding die afhankelijk is van de aanname φ_i en die vanaf stap $j+1$ is afgesloten. Daarom mag er vanaf stap $j+1$ ook aan geen van de van φ_i afhankelijke formules $\varphi_i, \varphi_{i+1}, \ldots, \varphi_j$ worden gerefereerd. Wel mag aldoor, zowel voor, in en na de sub-afleiding, naar eerdere, nog niet ingetrokken, formules (bijvoorbeeld φ_1) worden gerefereerd.

In een afleiding kunnen meerdere deelafleidingen voorkomen, echter alleen geheel na elkaar, als in Figuur 9.1(a), of genest, als Figuur 9.1(b). De kantlijnen mogen dus niet snijden, als in Figuur 9.1(c). Er geldt tenslotte nog de beperking dat alleen de laatst gemaakte nog geldige aanname mag worden ingetrokken; er mag er geen worden overgeslagen (zie Figuur 9.1(d)).

9.2 Regels

Volgens de boven gegeven uitleg kunnen afleidingsregels in drie soorten worden onderverdeeld:

1. Een formule opvoeren als aanname.
2. De regels die gepaard gaan met het intrekken van een aanname—het stukje afleiding dat van die aanname afhankelijk is—wordt door een extra kantlijn van de rest afgezonderd (I→ en I¬);
3. De overige, "gewone", regels, die kunnen worden toegepast om in een afleiding ψ te concluderen onder verwijzing naar formules $\varphi_1, \ldots \varphi_n$; deze worden hier weergegeven in het schema

$$\frac{\varphi_1, \ldots \varphi_n}{\psi}\ \textit{Rechtvaardiging}$$

Eerst vermelden we nu de gewone introductie- en gebruiksregels.

$$\frac{\varphi, \psi}{\varphi \wedge \psi}\ \mathrm{I}\wedge$$

$$\frac{\varphi \wedge \psi}{\varphi}\ \mathrm{G}\wedge \qquad \frac{\varphi \wedge \psi}{\psi}\ \mathrm{G}\wedge$$

$$\frac{\varphi}{\varphi \vee \psi}\ \mathrm{I}\vee \qquad \frac{\psi}{\varphi \vee \psi}\ \mathrm{I}\vee$$

$$\frac{\varphi_1 \vee \varphi_2,\ \varphi_1 \to \psi,\ \varphi_2 \to \psi}{\psi}\ \mathrm{G}\vee$$

$$\frac{\varphi,\ \varphi \to \psi}{\psi}\ \mathrm{G}\to \qquad \frac{\varphi,\ \neg\varphi}{\bot}\ \mathrm{G}\neg$$

Vervolgens nog twee introductieregels waarbij aannamen worden ingetrokken::

$i.$	φ	ass
$..$	$\ldots$	$\ldots$
$j.$	ψ	$\ldots$
$j+1.$	$\varphi \to \psi$	I$\to$, i–j

$i.$	φ	ass
$..$	$\ldots$	$\ldots$
$j.$	$\bot$	$\ldots$
$j+1.$	$\neg\varphi$	I$\neg$, i–j

Dan resteren er nog drie regels, die het patroon van introductie- en eliminatieregels doorbreken.

$$\frac{\bot}{\varphi}\ \bot\text{-regel} \qquad \frac{\neg\neg\varphi}{\varphi}\ \neg\neg\text{-regel} \qquad \frac{\varphi}{\varphi}\ \text{Herh}$$

Hier volgt een voorbeeld-afleiding.

1.	p	ass
2.	q	ass
3.	$p \wedge q$	I$\wedge$, 1, 2
4.	$(p \wedge q) \vee r$	I$\vee$, 3

Een tweede voorbeeld:

1.	$p \wedge q$	ass
2.	$q \to r$	ass
3.	q	G$\wedge$, 1
4.	r	G$\to$, 2, 3
5.	p	G$\wedge$, 1
6.	$p \wedge r$	I$\wedge$, 4, 5

Bij een bewijssysteem hoort een formeel begrip van *afleidbaarheid*. Dat kan nu worden gedefinieerd.

Definitie 9.1 i) *Voor een verzameling formules* Γ *schrijven we*

$$\Gamma \vdash \psi$$

(spreek uit: ψ *is afleidbaar uit* Γ*) wanneer er een afleiding is met conclusie* ψ *en met enkel niet-vervallen aannames uit* Γ.

ii) *Als* Γ *eindig is,* $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$*, dan schrijven we meestal*

$$\varphi_1, \ldots, \varphi_n \vdash \psi$$

in plaats van $\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$.

iii) *In het geval* $\Gamma = \emptyset$ *schrijven we* $\vdash \psi$ *in plaats van* $\Gamma \vdash \psi$*; de formule* ψ *heet dan gewoon afleidbaar.*

Om terug te komen op de voorbeeldafleidingen: de eerste toont aan dat

$$p, q \vdash (p \wedge q) \vee r,$$

en de tweede dat

$$p \wedge q, q \to r \vdash p \wedge r.$$

Maar uit de eerste afleiding volgt bijvoorbeeld ook dat

$$p, q, \neg s \vdash (p \wedge q) \vee r.$$

9.3 Uitleg van de regels

Met name het gebruik van de regels (G$\vee$), (I$\to$) en (I$\neg$) vergt nog wel enige toelichting. Ook de andere regels zullen hieronder worden besproken en gemotiveerd.

(I$\wedge$) Als φ en ψ beide aangetoond zijn, kan $\varphi \wedge \psi$ geconcludeerd worden.

(G$\wedge$) Uit $\varphi \wedge \psi$ kunnen φ en ψ geconcludeerd worden.

(I$\vee$) Men kan $\varphi \vee \psi$ concluderen als tenminste één van φ en ψ aangetoond zijn. (Mag ook allebei.)

(G$\vee$) Uit $\varphi_1 \vee \varphi_2$ kan men ψ concluderen, als zowel φ_1 als φ_2 afzonderlijk ψ impliceren. Deze regel is iets gecompliceerder dan de drie voorgaande. Merk op dat er verwantschap is met het redeneren met behulp van gevalonderscheiding: als je weet dat er maar twee gevallen zijn, A en B, en dat in beide gevallen C moet gelden, dan weet je dat C geldt.

(I$\to$) Om $\varphi \to \psi$ af te leiden wordt φ aangenomen met de bedoeling uitgaande daarvan ψ af te leiden. Lukt dit, dan wordt ψ kennelijk door φ geïmpliceerd. Er wordt tot $\varphi \to \psi$ geconcludeerd, waarbij de aanname φ kan worden ingetrokken. Alle formules in de sub-afleiding vanaf φ tot en met de laatste stap voor $\varphi \to \psi$ mogen na het intrekken van de aanname φ niet meer worden gebruikt, omdat zij immers zijn afgeleid onder de aanname φ.

De regel (I$\to$) mag alleen op de laatste aanname toegepast worden. Als dit niet het geval zou zijn, dan zouden er zomaar aannames kunnen verdwijnen, hetgeen niet de bedoeling is.

(G$\to$, Modus Ponens) Zodra φ bekend is, kan de formule $\varphi \to \psi$ gebruikt worden om ψ te concluderen.

(I$\neg$) Indien de aanname φ tot een contradictie leidt, dan geldt φ kennelijk niet en concludeer je tot $\neg\varphi$. De aanname kan (en moet) daarbij weer worden ingetrokken.

(G$\neg$) De formules φ en $\neg\varphi$ zijn met elkaar in tegenspraak; je kunt er dus $\bot$ uit concluderen.

NB. De negatie $\neg\varphi$ kan als de implicatie $\varphi \to \bot$ worden geïnterpreteerd. Dan corresponderen de introductie- en de gebruiksregel voor $\neg$ precies met (I$\to$) en met (G$\to$), en zijn ze dus eigenlijk overbodig. Om de correspondentie te benadrukken zijn de corresponderende regels in het overzicht naast elkaar gezet.

($\bot$-regel) Deze en de volgende regel hebben te maken met het speciale karakter van $\bot$ en $\neg$. In dit geval: *ex falso sequitur quod libet*. Vrij vertaald: “uit de contradictie is men vrij alles af te leiden”.

($\neg\neg$-regel) Men kan tot φ concluderen als $\neg\neg\varphi$ afgeleid is. Dit maakt bewijzen uit het ongerijmde mogelijk: teneinde φ te bewijzen, neem je eerst $\neg\varphi$ als aanname aan en probeer je onder die aanname $\bot$ af te leiden. Met (I$\neg$) volgt dan $\neg\neg\varphi$ en met de $\neg\neg$-regel tenslotte φ.

NB. Deze bewijsfiguur wordt niet door alle logici aanvaard. Er bestaat een stroming, het zg. *intuïtionisme*, geïnitieerd door de Nederlandse wiskundige L.E.J. Brouwer, die de $\neg\neg$-regel verwerpt. Een gevolg is dat het principe van de uitgesloten derde $\varphi \vee \neg\varphi$ dan niet meer kan worden afgeleid.

(Herh-regel) De herhalingsregel is opgenomen voor technische redenen. Een typisch voorbeeld van het gebruik van (Herh) is voorbeeld in het bewijs van de trivialiteit $\vdash p \to p$ hieronder. Strikt genomen is de regel echter overbodig. Alles wat met de herhalingsregel kan worden bewezen, kan dat, eventueel via een omweggetje, ook zonder.

We besluiten de behandeling van natuurlijke deductie nu met nog een paar voorbeelden en vervolgens een serie opgaven.

Voorbeeld voor (I$\to$). We willen laten zien dat $p, (p \wedge q) \to r \vdash q \to r$. Merk op dat in de gezochte afleiding zonder meer de aannames p en $(p \wedge q) \to r$ mogen worden gemaakt (stap 1 en 2). Vervolgens nemen we om $q \to r$ te bewijzen als extra aanname q aan (stap 3), met het voornemen om eerst r af te leiden (lukt in stap 5) en dan q met een toepassing van de I$\to$-regel weer in te trekken (stap 6). Nadat deze strategie is bepaald, is de afleiding eenvoudig af te maken:

1.	p	ass
2.	$p \wedge q \to r$	ass
3.	$\quad q$	ass
4.	$\quad p \wedge q$	I$\wedge$, 1, 3
5.	$\quad r$	G$\to$, 2, 4
6.	$q \to r$	I$\to$, 3–5

Voorbeeld voor (G$\vee$). We willen laten zien dat $p \vee q, q \to r \vdash p \vee r$. We starten natuurlijk weer met de aannames $p \vee q$ en $q \to r$ (stap 1 en 2). Om nu gebruik te kunnen maken van de G$\vee$-regel met $p \vee q$ als disjunctieve premisse, moeten we eerst de formules $p \to (p \vee r)$ en $q \to (p \vee r)$ zien te krijgen. Voor de eerste nemen we p als extra aanname aan (stap 3); we kunnen dan meteen al tot $p \vee q$ concluderen (stap 4) en de extra aanname p met een toepassing van (I$\to$) weer intrekken (stap 5). Met een soortgelijke sub-afleiding komen kunnen we vervolgens $q \to (p \vee r)$ bewijzen (stap 6-9). En tenslotte rest dan volgens plan nog de toepassing van (G$\vee$).

1.	$p \vee q$	ass
2.	$q \to r$	ass
3.	$\quad p$	ass
4.	$\quad p \vee r$	I$\vee$, 3
5.	$p \to (p \vee r)$	I$\to$, 3–4
6.	$\quad q$	ass
7.	$\quad r$	G$\to$, 2, 6
8.	$\quad p \vee r$	I$\vee$, 7
9.	$q \to (p \vee r)$	I$\to$, 6–8
10.	$p \vee r$	G$\vee$, 1, 5, 9

Voorbeelden voor (Herh). Het volgende voorbeeld van een toepassing van de herhalingsregel, een afleiding zonder aannames van de trivialiteit $p \to p$.

1.	$\quad p$	ass
2.	$\quad p$	Herh, 1
3.	$p \to p$	I$\to$, 1–2

Er geldt dus $\vdash p \to p$.

Het volgende voorbeeld van een toepassing van de herhalingsregel, een afleiding zonder aannames voor de tautologie $p \to (q \to p)$, toont aan dat $\vdash p \to (q \to p)$.

1.	$\quad p$	ass
2.	$\quad\quad q$	ass
3.	$\quad\quad p$	Herh, 1
4.	$\quad q \to p$	I$\to$, 2–3
5.	$p \to (q \to p)$	I$\to$, 1–4

Voorbeelden met negatie. Er wordt $\neg(p \wedge \neg p)$ afgeleid door de standaardstrategie voor het afleiden van een negatie te volgen: neem $p \wedge \neg p$ aan als aanname en probeer $\bot$ af te leiden; het resultaat volgt dan met (I$\neg$).

1.	$p \wedge \neg p$	ass
2.	$\quad p$	G$\wedge$, 1
3.	$\quad \neg p$	G$\wedge$, 1
4.	$\quad \bot$	G$\neg$, 2, 3
5.	$\neg(p \wedge \neg p)$	I$\neg$, 1–4

In het volgende voorbeeld wordt in stap 5 de $\bot$-regel

gebruikt om uit een contradictie q te concluderen.

1.	$p \vee q$	ass
2.	$\quad \neg p$	ass
3.	$\quad\quad p$	ass
4.	$\quad\quad \bot$	G$\neg$, 2, 3
5.	$\quad\quad q$	$\bot$-regel, 4
6.	$\quad p \to q$	I$\to$, 3–5
7.	$\quad\quad q$	ass
8.	$\quad\quad q$	Herh, 7
9.	$\quad q \to q$	I$\to$, 7–8
10.	$\quad q$	G$\vee$, 1, 6, 9
11.	$\neg p \to q$	I$\to$, 2–10

Tenslotte een voorbeeld waar we niet zonder de $\neg\neg$-regel kunnen. Om te laten zien dat $\neg p \to p \vdash p$, nemen we $\neg p \to p$ natuurlijk eerst als aanname. Dan kunnen we uit $\neg p$ falsum afleiden (stap 2-4). We concluderen tot $\neg\neg p$ en met de $\neg\neg$-regel tot p.

1.	$\neg p \to p$	ass
2.	$\quad \neg p$	ass
3.	$\quad p$	G$\to$, 1, 2
4.	$\quad \bot$	G$\neg$, 2, 3
5.	$\neg\neg p$	I$\neg$, 2–4
6.	p	$\neg\neg$-regel, 5

Opgave 9.1 Bewijs de volgende identiteiten met behulp van natuurlijke deductie.

1. (a) $p \wedge \neg r \vdash \neg r \wedge p$
   (b) $p \to q, p \to r \vdash p \to (q \wedge r)$
   (c) $p \to (p \to q) \vdash p \to q$
   (d) $(p \wedge \neg q) \to r, p \vdash \neg q \to (r \vee s)$
   (e) $(p \vee q) \to r \vdash q \to r$
   (f) $p \vee q \vdash (p \to q) \to q$
   (g) $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$
   (h) $p \to \neg q \vdash q \to \neg p$
   (i) $\vdash p \to \neg\neg p$
   (j) $p \vee q, \neg q \vdash \neg(p \to q)$
   (k) $\vdash \neg p \to (p \to q)$
   (l) $\neg p, \neg q \vdash \neg((p \to q) \to q)$
   (m) $\neg p \to q \vdash p \vee q$
   (n) $\vdash p \vee \neg p$
2. (a) $\neg(p \vee q) \vdash \neg p \wedge \neg q$
   (b) $p \wedge (q \wedge r) \vdash r \wedge p$
   (c) $p \to (q \wedge r), q \to s, p \vdash s$
   (d) $p \vee (p \wedge q) \vdash p$
   (e) $p \vdash p \vee (p \wedge q)$
   (f) $r \vdash r \wedge (r \vee s)$
   (g) $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$
   (h) $\vdash (p \to (q \to r)) \to ((p \to q) \to (p \to r))$
   (i) $p \to q, r \to s \vdash (p \vee r) \to (q \vee s)$
   (j) $\vdash ((p \to q) \wedge (r \to s)) \to ((p \vee r) \to (q \vee s))$
   (k) $p \to \neg p, \neg p \to p \vdash \bot$
   (l) $\neg(p \wedge q) \vdash \neg p \vee \neg q$
   (m) $\vdash \neg(p \to q) \to (p \wedge \neg q)$
   (n) $\vdash ((p \to q) \to p) \to p$ (Hint: probeer eerst $\neg\neg p$ af te leiden.)

Opgave 9.2 (Meta-logica.)

1. Bewijs dat voor elke formule uit de taal van de propositielogica, φ, het volgende geldt.

   Als
   $$\neg\varphi \vdash \varphi,$$
   dan
   $$\vdash \varphi.$$

   Aanwijzing: je wordt nu geacht *over* het systeem van natuurlijke deductie te redeneren. Je "staat" er nu dus buiten. In het bijzonder zal het bewijs dat je zult geven in je antwoord geen Fitch-bewijs zijn, maar een redenering over hoe het bestaan van een Fitch-bewijs voor φ uit $\neg\varphi$ het bestaan van een (ander) Fitch-bewijs voor φ impliceert.

2. Bewijs de twee volgende identiteiten.

   (a) Voor elke eindige verzameling van formules uit de propositielogica, Γ, en elke formule uit de propositielogica, φ, geldt het volgende:

   $\Gamma \cup \{\varphi\} \vdash \psi$ als en slechts als $\Gamma \vdash \varphi \to \psi$.

   Hint: de makkelijkste implicatie loopt van rechts naar links.

   (b) $\Gamma \vdash \varphi$ en $\Gamma \cup \{\varphi\} \vdash \psi$ impliceert $\Gamma \vdash \psi$.
   Geldt de omgekeerde implicatie ook? Zo ja, geef dan een bewijs. Zo nee, geef dan een (bij voorkeur zo eenvoudig mogelijk) tegenvoorbeeld.

9.4 Correctheid

Hier aangekomen heb je hopelijk een groot aantal bewijzen geconstrueerd, en goede intuïtie gekregen voor de werking van natuurlijke deductie. Zo niet, dan verwijzen we je naar Opdracht 9.1.

Het geven van vele bewijzen leidt op enig moment tot de vraag of natuurlijke deductie correct is. Is natuurlijke deductie bijvoorbeeld krachtig genoeg om er alle geldige stellingen mee te kunnen afleiden? En als dat zo mocht

zijn, is het dan niet per ongeluk té krachtig? I.e., kunnen we uitsluiten dat er met natuurlijke deductie per ongeluk onware stellingen worden afgeleid?

Deze vragen hebben betrekking op twee wenselijke eigenschappen. Laat Γ een eindige verzameling van formules, en φ een formule uit de propositielogica zijn.

– *Gezondheid.* Als $\Gamma \vdash \varphi$ dan ook $\Gamma \models \varphi$.
– *Volledigheid.* Als $\Gamma \models \varphi$ dan ook $\Gamma \vdash \varphi$.

Als een afleidingssysteem *volledig* is dan wil dat zeggen dat het krachtig genoeg is om alle ware stellingen te kunnen afleiden. Als een afleidingssysteem *gezond* is dan wil dat zeggen dat er geen rommel kan worden afgeleid. Een afleidingssysteem heet *correct* als en alleen als het gezond en volledig is. In een correct afleidingssysteem kunnen alle ware stellingen worden afgeleid, en is het onmogelijk om onware stellingen af te kunnen leiden.

Hoe bewijzen we nu dat natuurlijke deductie correct is? Om daar gevoel voor te krijgen is het allereerst belangrijk door te krijgen dat gezondheid in het algemeen makkelijk te bewijzen is.

Gezondheid

Om te laten zien dat natuurlijke deductie bewijzen gezond zijn hoef je eigenlijk alleen maar te laten zien dat alle afleidingsregels gezond zijn. Bewijzen zijn immers niets meer dan aaneenschakelingen van afleidingsregels.

Als voorbeeld bewijzen we de gezondheid van I$\vee$. We moeten dus laten zien dat voor alle formules φ, ψ het volgende geldt:

$$\varphi \models \varphi \vee \psi$$

Dat is eigenlijk meteen in te zien: zij m een willekeurig model, zó dat $m \models \varphi$. Dan per definitie van $\vee$ $m \models \varphi \vee \psi$. Klaar. Uiteraard analoog voor een verzwakking in het linkerlid: $\psi \models \varphi \vee \psi$.

Als tweede voorbeeld bewijzen we de gezondheid van I$\rightarrow$. We moeten dus laten zien dat voor alle formules φ, ψ het volgende geldt:

$$\text{Als } \varphi \models \psi, \text{ dan } \varphi \rightarrow \psi.$$

Dat is ook meteen na te gaan met modellen. Klaar.

Opgave 9.3 1. Bewijs de gezondheid van de G$\vee$-regel.

2. Bewijs de gezondheid van de $\bot$-regel.

Eigenlijk zij we nu klaar. Geloof je nu nog niet dat $\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$ bewezen is, stel dan dat $\Gamma \vdash \varphi$. Te bewijzen $\Gamma \models \varphi$. Stel $\Gamma \vdash_n \varphi$ vanwege een bewijs ter lengte $n \geq 1$. Als $n = 1$, dan moet φ een aanname zijn geweest en derhalve uit Γ komen. Dan zeker $\Gamma \models \varphi$ (ga na!). Neem aan dat we voor alle deelbewijzen hebben aangetoond dat deze sound zijn. (Zo niet, toon dan eerst de correctheid van deze deelbewijzen aan.) We gaan nu gevalsonderscheid naar de laatste stap van het gehele bewijs doen.

– De laatste stap is I$\vee$. Dus een kleiner deelbewijs bewijst zoiets als $\Gamma \vdash_k \varphi$, $k < n$, en het gehele bewijs bewijst $\Gamma \vdash_n \varphi \vee \psi$. Van een kleiner deelbewijs mogen we aannemen dat deze tot dusver gezond is: $\Gamma \models \varphi$. Maar daaruit volgt bijna meteen $\Gamma \models \varphi \vee \psi$. (Ga desnoods na met modellen.) Dus het gehele bewijs, inclusief de laatste stap, is ook gezond.

– ...

– ...

– De laatste stap is I$\rightarrow$. Dus een kleiner deelbewijs bewijst zoiets als $\Gamma \cup \{\varphi\} \vdash_k \psi$ (de aanname φ is nog niet ingetrokken) en het gehele bewijs bewijst $\Gamma \vdash_n \varphi \rightarrow \psi$. Weer mogen we aannemen dat het deelbewijs correct is, zodat $\Gamma \cup \{\varphi\} \models \psi$. Daaruit volgt vrijwel meteen $\Gamma \models \varphi \rightarrow \psi$.

– ...

– ...

Als alle gevallen zijn afgehandeld, zijn we klaar.

Volledigheid

Dit is andere koek. We moeten dan namelijk laten zien dat voor elke geldige semantische gevolgtrekking $\Gamma \models \varphi$ een bewijs van φ uit Γ bestaat. We hoeven overigens niet zelf met een bewijs op de proppen te komen of zelfs maar een bewijs aan te wijzen. Het is voldoende als we kunnen aantonen dat een Fitch bewijs met aannamen Γ en conclusie φ *bestaat*. (Dit heet een *existentieel bewijs*.) Maar zelfs het geven van een existentieel bewijs is hier nog niet zo makkelijk.

Hoe gaat men te werk? Er zijn tenminste drie manieren.

1. Bewijs direct $\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$. We verklappen meteen: dat gebeurt nooit. Hoe zouden we immers een bijbehorend bewijs moeten vinden?

2. Bewijs de contrapositie: $\Gamma \nvdash \varphi \Rightarrow \Gamma \not\models \varphi$. Immers, te beweren dat $\neg A$ volgt uit $\neg B$ is hetzelfde als te beweren dat B volgt uit A.

   Deze methode komt er dus op neer aan te tonen dat, als er geen Fitch-bewijs voor φ uit Γ bestaat, er een model m te fabriceren valt dat $\Gamma \cup \{\neg\varphi\}$ vervult. (Immers, $\Gamma \not\models \varphi \Leftrightarrow \Gamma \cup \{\neg\varphi\}$ is vervulbaar.)

3. Zoek een ander bewijssysteem waarvan we weten dat het compleet is, stel dit heet "H", en toon dan aan dat

$$\Gamma \vdash_{\mathrm{H}} \varphi \Rightarrow \Gamma \vdash_{\mathrm{Fitch}} \varphi.$$

De volledigheid van natuurlijke deductie bewijzen we later via methode (3). "H" staat dan voor Hilbert's systeem. Het kan ook goed via (2), maar het grappige is dat methode (2) in Hilbert iets makkelijker is.

9.5 Hilbert's systeem

In Hilbert's systeem wordt een aantal *axioma's* als uitgangspunt gekozen, en uit die axioma's worden met behulp van slechts één afleidingsregel, te weten *Modus Ponens* (MP), nieuwe formules afgeleid, de zogenaamde *stellingen*. Bij het afleiden van een conclusie φ uit een verzameling premissen $\{\psi_1,\ldots,\psi_n\}$ is het nu de kunst om, met behulp van de afleidingsregels, φ af te leiden uit de axioma's plus de premissenverzameling.

Een verzameling axioma's plus een verzameling afleidingsregels heet een *deductief systeem* (of: een *axiomatiek*, een *axiomatische calculus*, of kortweg een *calculus*). We geven een deductief systeem S voor de propositielogische taal $\mathcal{T}$.

Definitie 9.2 *Formules van de volgende vormen zijn* **axioma's** *van de propositielogica.*

1. $\varphi \to (\psi \to \varphi)$
2. $(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$
3. $(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$.

Niets anders is een axioma.

Merk op dat er oneindig veel axioma's zijn.

Opgave 9.4 Ga na waarom de volgende formules axioma's zijn:

1. $p \to (p \to p)$
2. $(p \to q) \to (p \to (p \to q))$
3. $(p \to (q \to p)) \to ((p \to q) \to (p \to p))$
4. $(p \to (q \to (p \to q))) \to ((p \to q) \to (p \to (p \to q)))$
5. $(\neg\neg p \to \neg(q \to r)) \to ((q \to r) \to \neg p)$.

We nemen aan dat het deductieve systeem slechts één enkele afleidingsregel kent. Die regel is *Modus Ponens*:

> *Als φ is afgeleid en $(\varphi \to \psi)$ is afgeleid, dan mag ook ψ worden afgeleid.*

We gaan nu een formele definitie geven van het begrip *afleiding* (Eng.: *derivation*) in een deductief systeem. Iets preciezer gezegd: we definiëren het begrip "afleiding in deductief systeem S uit formuleverzameling Σ". Informeel komt de definitie neer op het volgende: een afleiding uit een formuleverzameling Σ is een eindig geordend rijtje formules, waarbij geldt: elke formule in het rijtje is een axioma, of een formule uit Σ, of een formule die met behulp van een afleidingsregel is verkregen uit voorafgaande formules in het rijtje. Een afleiding binnen S waarbij geen extra premissen worden gebruikt noemen we een *stelling* van S.

We zullen straks gaan demonstreren hoe je *in* en *over* het deductieve systeem S kunt redeneren, maar eerst voeren we wat nieuwe notatie in. De notatie voor "φ is afleidbaar uit formuleverzameling Σ" wordt:

$$\Sigma \vdash \varphi.$$

Wanneer we te maken hebben met een eindige verzameling Σ die bestaat uit formules $\{\psi_1,\ldots,\psi_n\}$ wordt ook wel de volgende notatie gebruikt

$$\psi_1,\ldots,\psi_n \vdash \varphi$$

in plaats van:

$$\{\psi_1,\ldots,\psi_n\} \vdash \varphi.$$

De notatie voor "φ is een stelling van het systeem" wordt:

$$\vdash \varphi.$$

Hier is een voorbeeld van een heel simpele afleiding binnen het zojuist gepresenteerde deductieve systeem S; de afleiding laat zien dat de formule r afleidbaar is uit de twee premissen $q \to r$ en $(p \to (p \to p)) \to q$.

Stelling 9.1 *Er geldt:* $q \to r, (p \to (p \to p)) \to q \vdash r$.

Bewijs:

1	$p \to (p \to p)$	[ax 1]
2	$(p \to (p \to p)) \to q$	[premisse]
3	q	[MP uit 1 en 2]
4	$q \to r$	[premisse]
5	r	[MP uit 3 en 4].

Bij elke formule uit het rijtje is aangegeven hoe we eraan komen: de eerste formule is een axioma volgens schema 1; de tweede formule is element van de premissen-verzameling; de derde is door Modus Ponens verkregen uit de eerste en de tweede; de vierde is weer een premisse; en de vijfde, de uiteindelijke conclusie, is verkregen door Modus Ponens uit de derde en de vierde. □

Nog een voorbeeld, om te laten zien dat p afleidbaar is uit de twee premissen $\neg p \to \neg q$ en q.

Stelling 9.2 *Er geldt:* $\neg p \to \neg q, q \vdash p$.

Bewijs:

1	$\neg p \to \neg q$	[premisse]
2	$(\neg p \to \neg q) \to (q \to p)$	[ax 3]
3	$q \to p$	[MP uit 1 en 2]
4	q	[premisse]
5	p	[MP uit 3 en 4].

Hiermee is de stelling bewezen. □

Het begrip *afleiding binnen een deductief systeem* schreeuwt natuurlijk om een recursieve definitie. We definiëren recursief de verzameling $\mathrm{AFL}_{S,\Sigma}$ van alle afleidingen binnen deductief systeem S uit formule-verzameling Σ.

Definitie 9.3 *De verzameling $AFL_{S,\Sigma}$ van alle afleidingen binnen deductief systeem S uit formule-verzameling Σ:*

– *Als φ een axioma is van S of een lid van Σ, dan is $\langle\varphi\rangle$ een lid van $AFL_{S,\Sigma}$.*

– *Als $\langle\psi_1,\ldots,\psi_n\rangle$ een lid is van $AFL_{S,\Sigma}$, en φ is een axioma van S, een lid van Σ, of een formule die met behulp van een afleidingsregel uit S verkregen is uit formules in $\{\psi_1,\ldots,\psi_n\}$, dan is $\langle\psi_1,\ldots,\psi_n,\varphi\rangle$ ook een lid van $AFL_{S,\Sigma}$.*
– *Niets anders is een lid van $AFL_{S,\Sigma}$.*

De formules die voorkomen als laatste element van een lid van $\mathrm{AFL}_{S,\Sigma}$ heten *afleidbaar* in S uit Σ. We kunnen Σ hier zien als een verzameling waaruit de premissen voor de afleiding mogen worden gekozen. Als Σ eindig is, is Σ dus een verzameling $\{\psi_1,\ldots,\psi_n\}$ van formules . In het algemeen hoeft Σ echter geen eindige verzameling te zijn.

Een speciaal geval hebben we wanneer Σ gelijk is aan de lege verzameling.

Definitie 9.4 *De formules die in een deductief systeem S afleidbaar zijn uit de lege verzameling heten de* **stellingen** *(Eng.:* theorems*) van S.*

Met andere woorden: de stellingen van een deductief systeem S zijn alle formules die voorkomen als laatste element van zeker lid van $\mathrm{AFL}_{S,\emptyset}$. Merk op dat elk axioma van systeem S een stelling is van het systeem.

Definitie 9.5 *Een afleiding in S uit de lege verzameling heet een* **bewijs** *in S (Eng.:* proof *in S).*

De bovenstaande definities waren voor willekeurige deductieve systemen. Vanaf nu concentreren we ons weer op het deductieve systeem voor de propositielogica waarvoor we hierboven de axioma's en de redeneerregel hebben gegeven. Overigens zijn er meerdere deductieve systemen voor de propositielogica mogelijk die equivalent zijn met het systeem dat wij hier hebben gepresenteerd, in de zin dat ze precies dezelfde verzameling stellingen opleveren.

Hier is een heel simpel voorbeeld van een bewijs van een stelling binnen ons deductieve systeem voor de propositielogica:

Stelling 9.3 *Er geldt: $\vdash p \to (p \to p)$.*

Bewijs:

$$1 \quad p \to (p \to p) \quad \text{[ax 1].}$$

Deze afleiding toont aan dat $p \to (p \to p)$ een stelling is in het deductieve systeem. □

Het systeem dat hierboven gegeven is, gebruikt een minimum aan axioma's en redeneerregels. Het geven van een zo zuinig systeem heeft als voordeel dat redeneren *over* het systeem gemakkelijk is. Het nadeel is dat het leveren van afleidingen *binnen* het systeem in eerste instantie wat lastiger is dan bij een systeem met meerdere afleidingsregels. Je moet nu eerst extra redeneer-hulpmiddelen gaan fabriceren: iedere stelling die je hebt bewezen mag vanaf dat moment worden gebruikt als was het een axioma.

Opgave 9.5 Zij gegeven de premissenverzameling $\{\neg\neg q \to \neg\neg p, q\}$. Leid hieruit de formule p af. Met andere woorden: laat zien dat

$$\neg\neg q \to \neg\neg p, q \vdash p.$$

Opgave 9.6 Toon aan: $p, q, r, p \to (q \to (r \to s)) \vdash s$.

Opgave 9.7 Toon aan: $\vdash (p \to q) \to (p \to p)$.

Het zo precies uitspellen van de definitie van *afleiding* (en daarmee samenhangend: die van *stelling*) als we hierboven gedaan hebben lijkt misschien pedant of anderszins overdreven, maar een recursieve formulering van het begrip *afleiding* is van belang wanneer we gaan redeneren *over* ons deductieve systeem: de recursieve definitie van *afleiding* maakt het mogelijk inductieve bewijzen te gaan leveren van zogenaamde *metastellingen* (Eng.: metatheorems), stellingen *over* het deductieve systeem, die moeten worden onderscheiden van de stellingen *van* het systeem.

We geven nu eerst nog een voorbeeld van een redenering *binnen* het systeem. Een bewijs voor $\varphi \to \varphi$ gaat als volgt (schrik vooral niet):

Stelling 9.4 *In het systeem S voor de propositielogica geldt voor elke formule φ het volgende: $\vdash \varphi \to \varphi$.*

Bewijs:

$$\begin{array}{ll} 1 & \varphi \to ((\varphi \to \varphi) \to \varphi) \\ 2 & (\varphi \to ((\varphi \to \varphi) \to \varphi)) \\ & \qquad \to ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi)) \\ 3 & (\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi) \\ 4 & \varphi \to (\varphi \to \varphi) \\ 5 & \varphi \to \varphi \end{array}$$

Rechtvaardigingen:[2]

$$\begin{array}{ll} 1 & \text{[ax 1, met } (\varphi \to \varphi) \text{ voor } \psi\text{]} \\ 2 & \text{[ax 2, met } (\varphi \to \varphi) \text{ voor } \psi\text{]} \\ 3 & \text{[MP uit 1 en 2]} \\ 4 & \text{[ax 1, met } \varphi \text{ voor } \psi\text{]} \\ 5 & \text{[MP uit 3 en 4].} \end{array}$$

Hiermee is het bewijs geleverd. □

Opmerking: dit is eigenlijk geen bewijs in het systeem S, maar een bewijs-*stramien*; elke invulling van een formule voor φ levert een bewijs op (vandaar dat het verhaal voor elke formule φ opgaat). φ is weer een meta-variabele.

Als je toch geschrokken bent van dit bewijs: zo'n afleiding construeer je natuurlijk niet van voor naar achter, maar *andersom*. Je constateert eerst dat $\varphi \to \varphi$ zelf geen axioma is, maar $\varphi \to (\varphi \to \varphi)$ wel. Dus zijn we klaar als we $(\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$ kunnen aantonen. Maar dat volgt weer uit het axioma

$$(\varphi \to ((\varphi \to \varphi) \to \varphi)) \to ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$$

[2] Normaal worden rechtvaardigingen onmiddellijk achter de formules geschreven, maar het ontbrak ons hier aan ruimte.

plus het axioma

$$\varphi \to ((\varphi \to \varphi) \to \varphi)$$

en klaar.

Goed, dit was een laatste voorbeeld van een afleiding *binnen* het systeem. Nu een voorbeeld van redeneren *over* het systeem. We zullen een metastelling bewijzen, de zogenaamde *deductiestelling*. In het bewijs van deze stelling wordt gebruik gemaakt van *inductie*, en wel als volgt. We laten eerst zien dat een bepaalde eigenschap geldt voor afleidingen van lengte 1 (dat wil zeggen afleidingen die uit een enkele formule bestaan; de afleiding van $p \to (p \to p)$ die we boven hebben gegeven is een voorbeeld). Dit is de basisstap in het inductie-bewijs. Daarna laten we zien dat *als* de eigenschap geldt voor afleidingen waarvan de lengte niet groter is dan n, *dan* geldt hij ook voor afleidingen van lengte $n+1$. Dat is de inductie-stap. Omdat de inductie-redenering betrekking heeft op de lengte van de afleiding noemen we dit: *inductie naar de lengte van een afleiding*. In § 7.4 hebben we kennisgemaakt met *inductie naar de complexiteit van een formule*, in het bewijs dat de welgevormde formules van een propositielogische taal in standaardnotatie evenveel linker- als rechter-haakjes hebben.

Stelling 9.5 (Deductiestelling) *Als* $\varphi \vdash \psi$, *dan* $\vdash \varphi \to \psi$.

Bewijs: We gebruiken inductie *naar de lengte van de afleiding* van ψ uit φ, dat wil zeggen van het bewijs van $\varphi \vdash \psi$.

– **basisstap:** De lengte van de afleiding is 1, dus: ofwel ψ is een axioma, of $\psi = \varphi$. In het eerste geval is

1	ψ	[axioma volgens gegeven]
2	$\psi \to (\varphi \to \psi)$	[ax 1]
3	$\varphi \to \psi$	[MP uit 1 en 2].

de gevraagde afleiding van $\varphi \to \psi$, in het tweede geval moeten we $\varphi \to \varphi$ bewijzen. Dat hebben we in stelling 9.4 al gedaan, dus: klaar. Dit was het basisgeval.

– **inductiestap:** De inductiehypothese luidt: de bewering geldt voor alle paren φ, ψ waarbij de lengte van de afleiding van ψ uit φ niet groter is dan n. We bekijken nu een paar φ, ψ waarbij ψ in $n+1$ stappen uit φ is afgeleid. Er zijn drie mogelijkheden:

1. ψ is een axioma. Bewijs nu $\varphi \to \psi$ als in de basisstap.
2. $\psi = \varphi$. Bewijs $\varphi \to \psi$ als in de basisstap.
3. ψ komt via MP uit eerdere χ, $\chi \to \psi$ in de afleiding.

   In dit laatste geval is het recept voor de afleiding van $\varphi \to \psi$ als volgt. Merk eerst op dat op χ en $\chi \to \psi$ de inductiehypothese van toepassing is. We hebben dus: $\vdash \varphi \to \chi$ en $\vdash \varphi \to (\chi \to \psi)$. De afleiding voor $\vdash \varphi \to \psi$ is nu

$\vdots$	
$\varphi \to \chi$	[afleiding van $\varphi \to \chi$]
$\vdots$	
$\varphi \to (\chi \to \psi)$	[afleiding van $\varphi \to (\chi \to \psi)$]
$(\varphi \to (\chi \to \psi)) \to$	
$((\varphi \to \chi) \to (\varphi \to \psi))$	[ax 2]
$(\varphi \to \chi) \to (\varphi \to \psi)$	[MP]
$\varphi \to \psi$.	[nogmaals MP]

Dit is de gevraagde afleiding, en hiermee is de deductiestelling rond. □

Opgave 9.8 Lever een bewijs van het omgekeerde van de deductiestelling: "*als* $\vdash \varphi \to \psi$ *dan* $\varphi \vdash \psi$".

De deductiestelling kan het redeneren binnen het systeem gemakkelijker maken omdat we haar mogen gebruiken als extra afleidingsregel. Voordat we dat gaan demonstreren roepen we je op om zelf een versterking van de deductiestelling te bewijzen:

Opgave 9.9 Werk het bewijs van de deductiestelling uit tot een bewijs voor een meer algemene variant van deze stelling die het volgende zegt:

$$\text{Als } \varphi_1,\ldots,\varphi_n,\varphi_{n+1} \vdash \psi \text{ dan } \varphi_1,\ldots,\varphi_n \vdash (\varphi_{n+1} \to \psi).$$

We zullen de versterkte versie van de deductiestelling gebruiken om de volgende stelling te bewijzen.

Stelling 9.6 $\varphi \to \psi, \psi \to \chi \vdash \varphi \to \chi$.

Bewijs: Om dit te laten zien hoeven we volgens de deductiestelling (versie uit opdracht 9.9) alleen het volgende aan te tonen:

$$\varphi \to \psi, \psi \to \chi, \varphi \vdash \chi.$$

Dit is simpel:

1	φ	[gegeven]
2	$\varphi \to \psi$	[gegeven]
3	ψ	[MP uit 1 en 2]
4	$\psi \to \chi$	[gegeven]
5	χ	[MP uit 3 en 4].

Hiermee is de stelling bewezen. □

9.6 Beslisbaarheid, correctheid, volledigheid

Het zogenaamde *beslisbaarheidsprobleem* voor de propositielogica is de volgende vraag: bestaat er een mechanische procedure die, voor een willekeurige formule φ van de propositielogica, kan uitmaken of die formule een tautologie is? We hebben niet precies omschreven wat een mechanische procedure is, maar we mogen het beslisbaarheidsprobleem hier wel even als volgt parafraseren: valt er een C#-computerprogramma te

schrijven dat, nadat ik een willekeurige formule heb ingetikt, in een eindige hoeveelheid tijd kan uitmaken of die formule een tautologie is? Het antwoord is: ja. De waarheidstafelmethode en de semantische tableau-methode zijn methoden die met niet al te veel moeite zijn om te zetten in computerprogramma's: voor allebei deze werkwijzen geldt dat ze bestaan uit domweg regeltjes toepassen, en het gestelde probleem wordt beantwoord. Dus: de propositielogica is beslisbaar.

We komen nu toe aan de hamvraag die kan worden gesteld met betrekking tot het semantische perspectief en het afleidingsperspectief op propositielogisch redeneren: dekken de centrale begrippen uit het eerste perspectief (logisch gevolg) en die uit het tweede perspectief (afleidbaar uit) elkaar? Ofwel: dekken $\models$ en $\vdash$ elkaar? Deze vraag valt in twee onderdelen uiteen:

– **Ten eerste:** geldt *als* $\psi_1,\ldots,\psi_n \vdash \varphi$ *dan* $\psi_1,\ldots,\psi_n \models \varphi$?

Dit is de vraag naar de *correctheid* van het deductieve systeem. Een systeem is correct als geen enkele gevolgtrekking waarvoor in het systeem een afleiding kan worden gevonden logisch ongeldig blijkt te zijn.

Dat het in § 9.5 gegeven deductieve systeem voor de propositielogica correct is, is niet zo moeilijk te bewijzen. Gebruik waarheidstafels of semantische tableaus om te laten zien dat alle axioma's tautologieën zijn, en dus ook: logische gevolgen van elke premissenverzameling. Toon vervolgens aan dat Modus Ponens de eigenschap van logisch-gevolg-zijn van een gegeven premissenverzameling bewaart.

Opgave 9.10 Werk deze schets van het bewijs van de correctheid van ons deductief systeem voor de propositielogica uit.

Correctheid is heel handig wanneer we willen aantonen dat $\psi_1,\ldots,\psi_n \nvdash \varphi$. Het feit dat jij of iemand anders er niet in slaagt om een afleiding van φ uit $\psi_1,\ldots,\psi_n$ te vinden bewijst immers niet dat er niet zo'n afleiding bestaat ... Maar een tegenvoorbeeld voor $\psi_1,\ldots,\psi_n \models \varphi$ is in het algemeen snel gevonden. En *correctheid* zegt dan dat uit $\psi_1,\ldots,\psi_n \not\models \varphi$ volgt dat $\psi_1,\ldots,\psi_n \nvdash \varphi$.

Er is echter nog een tweede onderdeel van de vraag naar de verhouding tussen $\vdash$ en $\models$.

– **Ten tweede:** geldt *als* $\psi_1,\ldots,\psi_n \models \varphi$, *dan* $\psi_1,\ldots,\psi_n \vdash \varphi$?

Dit is de vraag naar de *volledigheid* van het deductieve systeem: is het deductieve systeem rijk genoeg om voor elke willekeurige logisch geldige gevolgtrekking een afleiding te kunnen produceren?

Een eerste kleine hobbel die we moeten nemen heeft te maken met het feit dat de axioma's in de boven gepresenteerde propositielogische calculus alleen gebruik maken van de voegwoorden $\neg$ en $\rightarrow$, terwijl de afleidingsregel Modus Ponens bij de axioma's aansluit door alleen gebruik te maken van het voegwoord $\rightarrow$. Één en ander betekent dat alleen formules kunnen worden afgeleid die alleen de voegwoorden $\neg$ en $\rightarrow$ bevatten. Het gebruik van alleen $\neg$ en $\rightarrow$ heeft voordelen voor het redeneren over de calculus: bij inductiebewijzen naar de complexiteit van formules hoeven we in de inductiestap maar twee gevallen te bekijken. Aan de andere kant is het handig om toch de beschikking te hebben over de andere voegwoorden, met name als we de brug tussen axiomatiek en semantiek willen slaan. De formules waarvan in de volledigheidsstelling sprake is kunnen immers de voegwoorden $\wedge$, $\vee$ en $\leftrightarrow$ bevatten.

Om hier een mouw aan te passen kunnen we formules die voorkomens van $\wedge$, $\vee$ en $\leftrightarrow$ bevatten beschouwen als *afkortingen* voor formules met alleen $\neg$ en $\rightarrow$. We spreken af:

– $\varphi \wedge \psi$ is een afkorting voor $\neg(\varphi \rightarrow \neg\psi)$.

– $\varphi \vee \psi$ is een afkorting voor $\neg\varphi \rightarrow \psi$.

– $\varphi \leftrightarrow \psi$ is een afkorting voor $\neg((\varphi \rightarrow \psi) \rightarrow \neg(\psi \rightarrow \varphi))$.

Je kunt gemakkelijk met behulp van waarheidstafels nagaan dat deze afkortingen de juiste betekenissen toekennen.

Ook nadat deze eerste hobbel is weggewerkt is de volledigheidsvraag veel moeilijker te beantwoorden dan de correctheidsvraag. De rest van deze paragraaf is eraan gewijd, maar wie de details bij eerste lezing wil overslaan heeft onze zegen. Wie tot volledig begrip van wat nu volgt wil geraken raden wij aan om de rest van de paragraaf niet alleen van voren naar achteren maar ook van achteren naar voren te lezen. Deze werkwijze maakt duidelijk wat de motivering is van de hulpstellingen die tot het uiteindelijke resultaat leiden.

We beginnen met het toepassen van *contrapositie*. Dit wil zeggen, we zetten de volledigheidsvraag om in:

– Geldt *als* $\psi_1,\ldots,\psi_n \not\vdash \varphi$, *dan* $\psi_1,\ldots,\psi_n \not\models \varphi$?

Om conclusies te kunnen trekken uit het feit dat een bepaalde formule *niet* afleidbaar is uit een gegeven stel premissen hebben we een flinke omweg nodig. Noem de premissenverzameling Γ. Wat we willen is een waardering vinden voor de formules van de taal die alle premissen waar maakt maar de conclusie φ onwaar. De weg die we zullen bewandelen om zo'n waardering te vinden is als volgt. We breiden de premissenverzameling eerst uit met de negatie van de conclusie. Dit geeft de verzameling $\Gamma \cup \{\neg\varphi\}$. Deze verzameling breiden we vervolgens verder uit tot een verzameling Γ^* van formules die de volgende eigenschap heeft: $\psi \in \Gamma^*$ desda $\Gamma^* \vdash \psi$. Wanneer dan ook nog uit de constructie van Γ^* volgt dat er een valuatie V is die precies de formules uit Γ^* waar maakt zijn we klaar. Die V zal de premissen uit de oorspronkelijke premissenverzameling waar maken en de conclusie onwaar, waarmee is aangetoond dat $\psi_1,\ldots,\psi_n \not\models \varphi$.

We voeren een paar nieuwe begrippen in. Allereerst het begrip *consistentie*.

Definitie 9.6 *Een verzameling formules* Γ *heet* **consistent** *wanneer er geen formule* φ *is waarvoor zowel* $\Gamma \vdash \varphi$ *als* $\Gamma \vdash \neg\varphi$.

Een voorbeeld van een consistente verzameling formules is $\{p, p \vee q, \neg q\}$. Een formuleverzameling die niet consistent is heet *inconsistent*; $\{p, \neg p\}$ is een voorbeeld van een inconsistente formuleverzameling.

Opgave 9.11 Laat zien: als een formuleverzameling Γ inconsistent is, dan is elke Γ' zo dat $\Gamma \subseteq \Gamma'$ eveneens inconsistent.

We bewijzen nu een paar stellingen die ons iets meer vertellen over het zojuist gedefinieerde begrip *consistentie*.

Stelling 9.7 *Een formuleverzameling Γ is consistent desda er een formule φ te vinden is zodat $\Gamma \nvdash \varphi$.*

Bewijs: De stelling bestaat uit twee onderdelen. Ten eerste: als Γ consistent is, dan is er een φ zodat $\Gamma \nvdash \varphi$. Ten tweede: als $\Gamma \nvdash \varphi$ dan is Γ consistent.

Het eerste is gemakkelijk. Als Γ consistent is, dan is er per definitie geen φ te vinden zo dat zowel φ zelf als zijn negatie afleidbaar is uit Γ. Neem dus een willekeurige formule φ. Ofwel $\Gamma \nvdash \varphi$, en klaar, ofwel: $\Gamma \vdash \varphi$ en daaruit volgt dat $\Gamma \nvdash \neg\varphi$, en ook klaar.

Nu de andere richting. Stel er is een φ zodat $\Gamma \nvdash \varphi$. We moeten nu laten zien dat Γ consistent is, dat wil zeggen dat er geen ψ is waarvoor $\Gamma \vdash \psi$ en $\Gamma \vdash \neg\psi$. Veronderstel even dat er wel zo'n ψ is: we hebben dan $\Gamma \vdash \psi$ en $\Gamma \vdash \neg\psi$. Als we nu kunnen laten zien dat $\Gamma \vdash \neg\psi \to (\psi \to \varphi)$, dan hebben we ook $\Gamma \vdash \varphi$ (tweemaal Modus Ponens), en dus een tegenspraak met het gegeven dat $\Gamma \nvdash \varphi$. We mogen dan concluderen dat de aanname dat Γ inconsistent is niet klopt, en we zijn klaar. (Dit is weer een *reductio ad absurdum*.)

We laten nu zien dat

$$\vdash \neg\psi \to (\psi \to \varphi)$$

en dus zeker:

$$\Gamma \vdash \neg\psi \to (\psi \to \varphi).$$

In stelling 9.6 hebben we bewezen dat

$$\varphi \to \psi, \psi \to \chi \vdash \varphi \to \chi.$$

We gebruiken dit hulpresultaat in de verlangde afleiding:

1	$\neg\psi \to (\neg\varphi \to \neg\psi)$	[ax 1]
2	$(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$	[ax 2]
3	$\neg\psi \to (\psi \to \varphi)$	[1, 2, hulpresultaat].

En klaar. Hiermee is de stelling bewezen. □

Opgave 9.12 Laat met behulp van stelling 9.7 zien dat de verzameling van alle propositieletters van een propositielogische taal $\mathcal{T}$ consistent is.

Stelling 9.8 *Als $\Gamma \nvdash \varphi$ dan is $\Gamma \cup \{\neg\varphi\}$ consistent.*

Bewijs: Neem aan dat $\Gamma \nvdash \varphi$. Dan is Γ kennelijk consistent (stelling 9.7). We doen nu weer een *reductio ad absurdum*. Veronderstel dat $\Gamma \cup \{\neg\varphi\}$ inconsistent is. Dan volgt uit stelling 9.7 dat

$$\Gamma \cup \{\neg\varphi\} \vdash \neg\alpha$$

voor een willekeurig propositielogisch axioma α. Toepassen van de deductiestelling levert nu: $\Gamma \vdash \neg\varphi \to \neg\alpha$. Hieruit volgt, met behulp van axiomaschema 3: $\Gamma \vdash \alpha \to \varphi$. Maar α was een axioma, dus we hebben zeker $\Gamma \vdash \alpha$, zodat we met Modus Ponens kunnen concluderen: $\Gamma \vdash \varphi$. Dit is in tegenspraak met het gegeven $\Gamma \nvdash \varphi$. Hiermee is de veronderstelling dat $\Gamma \cup \{\neg\varphi\}$ inconsistent is weerlegd. □

Definitie 9.7 *We noemen een formuleverzameling Γ* **maximaal consistent** *wanneer Γ de volgende twee eigenschappen heeft:*

- *Γ is consistent.*
- *Als $\Gamma \subset \Gamma'$, dan is Γ' inconsistent.*

Een maximaal consistente verzameling formules is een consistente verzameling formules waaraan geen formules meer kunnen worden toegevoegd zonder de verzameling inconsistent te maken.

Opgave 9.13 Laat zien: als V een waardering is voor de formules uit $\mathcal{T}$, dan is $\{\varphi \in \mathcal{T} \mid V(\varphi) = 1\}$ een maximaal consistente verzameling.

Stelling 9.9 (Lemma van Lindenbaum) *Elke consistente verzameling Γ van propositielogische formules is bevat in een maximaal consistente formuleverzameling Γ^*.*

Bewijs: Elke propositielogische taal $\mathcal{T}$ heeft aftelbaar veel formules. We mogen daarom aannemen dat we beschikken over een aftelling

$$\varphi_0, \varphi_1, \varphi_2, \varphi_3, \ldots$$

van alle formules van de taal. We definiëren nu, uitgaande van Γ, een rij van verzamelingen Γ_i met de eigenschap dat voor alle i: $\Gamma_i \subseteq \Gamma_{i+1}$. De definitie gaat als volgt:

- $\Gamma_0 = \Gamma$.
- $\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{\varphi_i\} & \text{als } \Gamma_i \cup \{\varphi_i\} \text{ consistent is.} \\ \Gamma_i & \text{anders.} \end{cases}$

Tenslotte nemen we de limiet van deze rij door al deze formuleverzamelingen op één hoop te vegen:

- $\Gamma^* = \bigcup_i \Gamma_i$.

Door inductie naar i is nu gemakkelijk in te zien dat elk van de verzamelingen Γ_i consistent is. Maar dan is ook Γ^* consistent. Stel immers van niet. Dan is er een formule ψ zo dat zowel $\Gamma^* \vdash \psi$ als $\Gamma^* \vdash \neg\psi$. Elk van deze afleidingen gebruikt slechts eindig veel premissen. Dus is er een getal m zo dat alle premissen die in een van beide afleidingen gebruikt worden in Γ_m zitten. Maar dan hebben we: $\Gamma_m \vdash \psi$ en $\Gamma_m \vdash \neg\psi$, met andere woorden Γ_m is inconsistent: een tegenspraak met wat we reeds hadden aangetoond.

Tenslotte geldt dat Γ^* maximaal consistent is. Neem een willekeurige formule ψ zo dat $\psi \notin \Gamma^*$ en beschouw de verzameling $\Gamma^* \cup \{\psi\}$. Formule ψ zit in onze aftelling van de formules van de taal, dus er is een k zo dat $\psi = \varphi_k$. Beschouw nu de stap in het constructieproces van Γ^* waar Γ_{k+1} werd gevormd. We weten dat $\varphi_k \notin \Gamma^*$, en dat kan alleen betekenen dat toevoeging van φ_k aan Γ_k de verzameling inconsistent zou hebben gemaakt. Maar dan is $\Gamma^* \cup \{\varphi_k\}$ eveneens inconsistent, want $\Gamma_k \subseteq \Gamma^*$. □

We bewijzen nu dat maximaal consistente verzamelingen de eigenschappen hebben die nodig zijn voor het volledigheidsresultaat.

Stelling 9.10 *Als Γ een maximaal consistente formuleverzameling is dan geldt voor elke formule φ: $\varphi \in \Gamma$ desda $\Gamma \vdash \varphi$.*

Bewijs: Uiteraard geldt dat als $\varphi \in \Gamma$ dan $\Gamma \vdash \varphi$. Om het omgekeerde te bewijzen nemen we aan dat $\varphi \notin \Gamma$. Uit de maximaliteit van Γ volgt dat $\Gamma \cup \{\varphi\}$ inconsistent is. Als nu $\Gamma \vdash \varphi$ dan volgt dat ook Γ inconsistent is, in tegenspraak met het gegeven omtrent Γ. Dus $\Gamma \nvdash \varphi$. □

Stelling 9.11 *Als Γ een maximaal consistente formuleverzameling is dan geldt voor elke formule φ: $\varphi \in \Gamma$ of $\neg\varphi \in \Gamma$.*

Bewijs: Stel voor een willekeurige formule φ dat $\neg\varphi \notin \Gamma$. Omdat Γ *maximaal* consistent is moet dus $\Gamma \cup \{\neg\varphi\}$ inconsistent zijn. Vanwege stelling 9.8 is derhalve $\Gamma \vdash \varphi$. Met stelling 9.10 volgt dat dan $\varphi \in \Gamma$. □

Merk op dat we deze stelling ook anders mogen formuleren: als Γ maximaal consistent is geldt dat $\varphi \in \Gamma$ desda $\neg\varphi \notin \Gamma$.

Stelling 9.12 *Als Γ een maximaal consistente formuleverzameling is, dan geldt: $\varphi \to \psi \in \Gamma$ desda (als $\varphi \in \Gamma$ dan $\psi \in \Gamma$).*

Bewijs: Neem aan dat Γ maximaal consistent is en dat $\varphi \to \psi \in \Gamma$ en $\varphi \in \Gamma$. Eenmaal toepassen van Modus Ponens geeft: $\Gamma \vdash \psi$, en met stelling 9.10 kunnen we concluderen dat $\psi \in \Gamma$.

Omgekeerd, neem aan dat Γ maximaal consistent is, en dat we weten dat voor een tweetal formules φ, ψ geldt: als $\varphi \in \Gamma$ dan $\psi \in \Gamma$. Twee mogelijkheden: ofwel ψ zit in Γ, en dan hebben we $\Gamma \vdash \varphi \to \psi$ (gebruik axioma $\psi \to (\varphi \to \psi)$ plus Modus Ponens), en dus $\varphi \to \psi \in \Gamma$ met stelling 9.10, ofwel φ, ψ geen van beide in Γ, en dan hebben we $\neg\varphi \in \Gamma$ met stelling 9.11, en opnieuw: $\Gamma \vdash \varphi \to \psi$ (gebruik axioma $\neg\varphi \to (\neg\psi \to \neg\varphi)$, pas Modus Ponens toe, en gebruik axiomaschema 3 plus nogmaals Modus Ponens om $\varphi \to \psi$ af te leiden), zodat ook in dit geval $\varphi \to \psi \in \Gamma$. □

We weten nu genoeg om te kunnen laten zien dat maximaal consistente verzamelingen geschikt zijn om de brug te slaan tussen axiomatiek en semantiek.

Stelling 9.13 *Als Γ een consistente verzameling formules is, dan is er een waardering V zo dat voor elke $\varphi \in \Gamma$: $V(\varphi) = 1$.*

Bewijs: Als Γ consistent is dan is er, zoals we hierboven hebben bewezen, een maximaal consistente Γ^* zo dat $\Gamma \subseteq \Gamma^*$. Neem nu de waardering V die gebaseerd is op de volgende interpretatie I voor de propositieletters van de taal $\mathscr{T}$: $I(p) = 1$ desda $p \in \Gamma^*$.

We laten zien dat voor de aldus gedefinieerde V geldt: $V(\varphi) = 1$ desda $\varphi \in \Gamma^*$. We gebruiken inductie naar de complexiteit van de formule φ.

- Basisstap: wanneer φ atomair geldt de bewering per definitie.
- Inductiestap:
  - Stel dat φ de vorm $\neg\psi$ heeft. Dan hebben we: $V(\varphi) = 1$ desda $V(\psi) = 0$ desda (inductiehypothese) $\psi \notin \Gamma^*$ desda (stelling 9.11) $\varphi \in \Gamma$.
  - Als φ de vorm $\psi \to \chi$ heeft hebben we: $V(\varphi) = 0$ desda ($V(\psi) = 1$ en $V(\chi) = 0$) desda (inductiehypothese) ($\psi \in \Gamma^*$ en $\chi \notin \Gamma^*$) desda (stelling 9.12) $\varphi \notin \Gamma^*$.

Omdat $\Gamma \subseteq \Gamma^*$ hebben we: als $\varphi \in \Gamma$ dan $V(\varphi) = 1$. □

De volledigheidsstelling volgt nu onmiddellijk:

Stelling 9.14 (Volledigheid van de propositielogica) *Als $\Gamma \nvdash \varphi$, dan $\Gamma \not\models \varphi$.*

Bewijs: Neem aan dat $\Gamma \nvdash \varphi$. Uit het feit dat formule φ niet uit Γ afleidbaar is volgt dat Γ consistent is, en $\Gamma \cup \{\neg\varphi\}$ ook. Volgens stelling 9.13 is er nu een waardering V die alle formules in $\Gamma \cup \{\neg\varphi\}$ waar maakt. Deze waardering V levert een tegenvoorbeeld waarin alle formules uit Γ waar zijn terwijl φ niet waar is. □

De volledigheidsstelling voor de propositielogica werd in 1920 bewezen door de logicus E. Post. De methode die wij hebben gevolgd is van L. Henkin.

9.7 Volledigheid van natuurlijke deductie

De volledigheid van natuurlijke deductie moet nog worden bewezen. Nu bekend is dat Hilbert's systeem volledig is, kan dat gemakkelijk worden gedaan volgens methode 3 uiteengezet op blz. 104.

Opgave 9.14

1. Laat zien dat alle instanties van axiomaschema's van Hilbert's systeem met natuurlijke deductie zijn af te leiden.
2. Laat zien dat alle afleidingsregels van Hilbert's systeem kunnen worden gesimuleerd met natuurlijke deductie.
3. Beredeneer dat alles wat kan worden afgeleid uit Γ met Hilbert, ook kan worden afgeleid uit Γ met Fitch.
4. Beredeneer dat natuurlijke deductie volledig is.

Opgave 9.15 Veronderstel het omgekeerde scenario, i.e., veronderstel dat we weten dat natuurlijke deductie volledig is.

1. Formuleer een aanpak om, uitgaande van de volledigheid van natuurlijke deductie, de volledigheid van Hilbert's systeem bewijzen.
2. Bediscussieer de haalbaarheid van een dergelijke aanpak, en licht de haalbaarheid toe met enkele voorbeelden, bijvoorbeeld door de relevantie en de bewijscomplexiteit van

$$\varphi, \psi \vdash_{\text{Hilbert}} \neg(\varphi \to \neg\psi)$$

te bespreken.

9.8 Variaties en uitbreidingen

We besluiten dit hoofdstuk met een paar opmerkingen over variaties op en uitbreidingsmogelijkheden van de propositielogica. In zekere zin is de predikatenlogica (zie volgend hoofdstuk) een uitbreiding van de propositielogica. De predikatenlogica gebruikt dezelfde waarheidsfunctionele connectieven als de propositielogica, en kan dus alle werk aan waar de propositielogica voor geschikt is, en met gemak.

In plaats van de voegwoord-analyse uit de propositielogica over te nemen (wat in de predikatenlogica gebeurt) zouden we ook kunnen overwegen die voegwoord-analyse te *herzien*. We stippen alleen een paar mogelijkheden aan.

We zouden naast de waarheidswaarden 1 en 0 ook een waarde # voor 'onbepaald' kunnen toelaten. Als we # lezen als 'nog niet bekend', dan liggen de volgende waarheidstafels voor de hand:

	$\neg$
1	0
0	1
#	#

$\wedge$	1	0	#
1	1	0	#
0	0	0	0
#	#	0	#

$\vee$	1	0	#
1	1	1	1
0	1	0	#
#	1	#	#

$\to$	1	0	#
1	1	0	#
0	1	1	1
#	1	#	#

$\leftrightarrow$	1	0	#
1	1	0	#
0	0	1	#
#	#	#	#

Dit is echter niet de enige interpretatiemogelijkheid voor #. We zouden # ook kunnen opvatten als 'onzinnig'. Dit levert de volgende tafels (een onzinnig onderdeel maakt de hele formule onzinnig):

	$\neg$
1	0
0	1
#	#

$\wedge$	1	0	#
1	1	0	#
0	0	0	#
#	#	#	#

$\vee$	1	0	#
1	1	1	#
0	1	0	#
#	#	#	#

$\to$	1	0	#
1	1	0	#
0	1	1	#
#	#	#	#

$\leftrightarrow$	1	0	#
1	1	0	#
0	0	1	#
#	#	#	#

Een propositielogisch systeem dat naast 1 en 0 ook nog een derde waarde # toelaat heet een *driewaardige propositielogica*. Driewaardige logica's worden wel gebruikt om het zogenaamde *presuppositie*-begrip—waar linguïsten in geïnteresseerd zijn—te behandelen. Zie voor meer informatie: [Gamut 1982], deel 1, hoofdstuk 5.

Bij de overstap van een tweewaardig naar een driewaardig systeem wordt de eis van functionele volledigheid voor stellen voegwoorden zwaarder. Immers, we moeten nu alle mogelijke functies van $\{0,1,\#\}^n$ naar $\{0,1,\#\}$ kunnen uitdrukken. Als je wilt kun je zelf nagaan dat het stel $\{\neg,\wedge,\vee,\to,\leftrightarrow\}$ onder geen van beide driewaardige lezingen die we zojuist hebben gegeven functioneel volledig is.

De overgang van een tweewaardig naar een meerwaardig propositielogisch systeem komt neer op een variatie waarbij gesleuteld wordt aan de manier waarop de zinsoperatoren worden opgevat. We kunnen ook besluiten nieuwe zinsoperatoren toe te voegen. Zo kan de stap worden gezet van gewone propositielogica naar propositionele *tijds*logica, naar *modale* propositielogica, of naar een combinatie van beide.

Modale propositielogica ontstaat door aan de gewone propositielogica operatoren $\Box$ ('het is noodzakelijk dat') en $\Diamond$ ('het is mogelijk dat') toe te voegen. We kunnen nu formules maken als de volgende:

- $\Box p$
- $\neg\Diamond\neg p$
- $\Box p \to p$
- $p \to \Diamond p$.

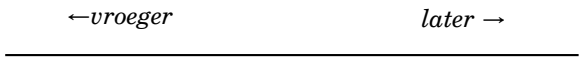
De intuïtie achter de nieuwe operatoren is: we willen *noodzakelijkheid* uitleggen op de manier van de filosoof Georg Wilhelm Leibniz (1646–1716). Leibniz legde uit: iets is noodzakelijk waar als het waar is *in alle mogelijke werelden*. In termen van de propositielogische semantiek:

Definitie 9.8 $\Box\varphi$ *is* **waar** *wanneer* φ *waar is in alle werelden* w*, waarbij een wereld* w *niets anders is dan een klassieke interpretatie voor de propositielogische variabelen.*

Een interpretatie voor modale predikatenlogica heeft dus een verzameling W van mogelijke werelden of contexten van node, en dient in *elke* wereld een waarheidswaarde toe te kennen aan alle propositieletters. Een *paar* bestaande uit een werelden-verzameling W en een interpretatiefunctie I die voor elke wereld een

waarheidswaarden-toekenning voor de variabelen oplevert heet een *Kripke-model* (naar de logicus Saul Kripke die zulke modellen in de jaren 50 voor het eerst voorstelde). Zie [Gamut 1982], deel 2, voor meer informatie.

De uitbreiding naar propositionele *tijds*logica gaat geheel analoog. Voeg tijdsoperatoren P en F toe aan de verzameling gewone propositielogische connectieven. $P\varphi$ staat nu voor: 'het is het geval geweest—minstens eens in het verleden—dat φ'. $F\varphi$ staat voor 'φ zal minstens eens vanaf nu het geval zijn'. Voor het interpreteren van formules uit de propositionele tijdslogica heb je een Kripke-model nodig waarbij de werelden gevormd worden door tijdstippen. Die tijdstippen kunnen *lineair* ten opzichte van elkaar geordend zijn, dat wil zeggen: ze kunnen als volgt op een as liggen:



De relatie tussen de tijdstippen is uiteraard de *eerder dan* relatie. Voor elk tijdstip t is er nu een waarheidswaarden-toekenning aan de propositionele variabelen. Weer verwijzen we naar [Gamut 1982], deel 2, en de daar genoemde literatuur voor meer informatie.

Deel III

Predikatenlogica

Hoofdstuk 10

Predikatenlogica

10.1 Kwantoren en variabelen

We zullen nu de propositielogica uitbreiden tot een meer verfijnd instrument: de predikatenlogica. De propositielogica is voor beperkte toepassingen heel adequaat, maar ze kan niet spreken over objecten (bv. getallen) in een zeker domein (bv. $\mathbb{Z}$) en de relaties tussen die objecten (bv. "$\leq$").

De *predikatenlogica* of PL—ook wel *predikatencalculus*, *elementaire logica*, *klassieke logica* of *1e-orde logica* genoemd—is ontwikkeld om naast de analyse van voegwoorden en de negatie-operator (alles op zinsniveau) ook de analyse van de *interne structuur* van zinnen ter hand te kunnen nemen, en zo te kunnen spreken over individuen in een zeker domein en de relaties tussen die individuen. Logisch gezien is dat van belang, want we willen graag iets kunnen zeggen over de geldigheid van redeneringen als de volgende:

Alle informatica-studenten eten drop.
Piet is een informatica-student.

Piet eet drop.

Met de propositielogica kunnen we hier niet volstaan, want propositielogisch kunnen we deze redenering alleen analyseren als:

$$\frac{p \quad q}{r}$$

en dat voldoet niet.

Nee, de logische geldigheid van bovenstaande redenering is wezenlijk gebaseerd op de rol van de operator *alle*. Een woord als *alle* noemen we een *kwantor*. Een zeer elegante theorie voor de behandeling van kwantoren is geleverd door de reeds eerder genoemde logicus Gottlob Frege: de moderne predikatenlogica is min of meer zijn geesteskind.

Frege merkte op dat bepaalde zinsdelen in een zin kunnen worden beschouwd als *functies*. Het functioneel karakter van die zinsdelen blijkt door een bepaald woord of een woordgroep in de zin te variëren en de rest constant te houden. Neem de zin "Jan slaat Piet". Haal de eigennaam *Jan* weg, en je krijgt "— slaat Piet". Ervan uitgaande dat de interpretatie van een zin een waarheidswaarde is, is de interpretatie van het zinsdeel "— slaat Piet" een karakteristieke functie: afhankelijk van de eigennaam die je invult krijg je de waarde 0 of 1 als uitkomst. Bij voorbeeld: voor *Marie* wordt "Marie slaat Piet" onwaar, voor "Jaap"wordt "Jaap slaat Piet" waar, enzovoorts. Maar we hebben in § 3.5 gezien dat zo'n karakteristieke functie eigenlijk niets anders is dan een *deelverzameling* van een of andere verzameling, in dit geval: een deelverzameling van de verzameling individuen die in aanmerking komen om onderwerp van gesprek te zijn. In jargon heet deze verzameling: het *discussiedomein* (Eng.: *domain of discourse*). Met andere woorden: "— slaat Piet" wordt geïnterpreteerd als de verzameling van alle individuen die Piet slaan. Net zo voor "Jan slaat —". Ook dit zinsdeel heeft een (karakteristieke functie van een) deelverzameling van het discussiedomein als interpretatie: de verzameling van alle individuen die door Jan geslagen worden.

Er is geen reden om hier op te houden: we kunnen ook nog de nog overblijvende eigennaam verwijderen uit het zinsdeel "Jan slaat —". Resultaat: "$—_1$ slaat $—_2$". De open plaatsen zijn genummerd om ze uit elkaar te kunnen houden. Dat dit nummeren nodig is blijkt bij voorbeeld wanneer we "Jan slaat zichzelf" en "Jan slaat Piet" vergelijken. De zinnen zijn verschillend van structuur, en het verschil kunnen we uitdrukken door te zeggen dat "Jan slaat zichzelf" gevormd is door de eigennaam *Jan* te combineren met "$—_1$ slaat $—_1$", terwijl "Jan slaat Piet" ontstaat door combinatie van *Jan* en "$—_1$ slaat Piet", een zinsdeel dat op zijn beurt is opgebouwd uit "$—_1$ slaat $—_2$", gecombineerd met *Piet*. Hieruit blijkt dat "$—_1$ slaat $—_1$" en "$—_1$ slaat $—_2$" uit elkaar moeten worden gehouden.

Hoe moeten we "$—_1$ slaat $—_2$" nu interpreteren? Wie zich § 3.1 nog herinnert zal niet verbaasd staan over het antwoord. De interpretatie van "$—_1$ slaat $—_2$" is een tweeplaatsige relatie op het discussiedomein. In 3.1 hebben we uitgelegd dat tweeplaatsige relaties op een verzameling A niets anders zijn dan deelverzamelingen van A^2, ofwel: verzamelingen van paren elementen uit A. In het geval van ons voorbeeld wordt dit een verzameling van paren elementen uit het discussiedomein zo dat telkens het eerste lid van het paar het tweede slaat, dus bij voorbeeld:

$$\{\langle \text{Jan}, \text{Piet}\rangle, \langle \text{Jan}, \text{Marie}\rangle, \langle \text{Annie}, \text{Wim}\rangle\}.$$

Weer is er geen reden om hier op te houden, bij zinsdelen die geïnterpreteerd worden als tweeplaatsige relaties op het discussiedomein. Uit "Haddock geeft Bobbie aan Kuifje" kunnen we, door verwijderen van eigennamen, de volgende functie verkrijgen:

$$—_1 \textit{ geeft } —_2 \textit{ aan } —_3$$

De interpretatie: een drieplaatsige relatie op het discussiedomein.

Merk nu op dat we het functionele zinsdeel "$—_1$ slaat Piet" niet alleen met een eigennaam kunnen combineren tot een zin, maar ook met behulp van een zogenaamde

kwantificerende uitdrukking, zoals *iemand*, *iedereen* of *niemand*: “Iedereen slaat Piet”, “Niemand slaat Piet”.

In het voorstel van Frege werd “—$_1$ slaat Piet” geïnterpreteerd als de verzameling van individuen in het discussiedomein die Piet slaan. *Jan* wordt geïnterpreteerd als een element in het discussiedomein, en de zin die ontstaat door *Jan* te combineren met “—$_1$ slaat Piet” is waar desda het element uit het discussiedomein dat de interpretatie is van *Jan* (de *referent* van *Jan*, om weer wat jargon te laten vallen) een element is van de interpretatie van “—$_1$ slaat Piet”.

Eigennamen worden geïnterpreteerd als elementen van het discussiedomein, maar kwantificerende uitdrukkingen gedragen zich anders dan eigennamen. Zo heeft “Niemand slaat Piet”, anders dan bij voorbeeld “Jan slaat Piet”, niet tot gevolg dat Jan geslagen wordt. Een ander logisch verschil tussen eigennamen en kwantificerende uitdrukkingen is dat

Jan slaat Piet.
Jan is iemand.

Iemand slaat Piet.

een geldige redenering is, waar

Iemand slaat Piet.
Iemand is Jan.

Jan slaat Piet.

niet geldig is. Dus kunnen we kwantificerende uitdrukkingen niet interpreteren als elementen van het discussiedomein. Frege’s oplossing maakt gebruik van variabelen, die we nu gaan introduceren.

De notatie met de genummerde open plaatsen is lastig. In plaats daarvan gebruiken we, in het voetspoor van Frege, uitdrukkingen die voor een willekeurig individu kunnen staan, de zogenaamde *individuele variabelen*. “—$_1$ slaat —$_1$” wordt nu “x slaat x”, en “—$_1$ slaat—$_2$” wordt “x slaat y”.

Zinnen als “Jan slaat Piet” en “Jan veracht zichzelf” kunnen nu wat betreft hun betekenis worden beschouwd als resultaten van de *uniforme substitutie* (*substitueren* is logisch jargon voor *vervangen*) van een naam voor een variabele, in een uitdrukking die die variabele bevat. Voorbeeld: uniforme substitutie van de naam *Jan* voor de variabele x in de uitdrukking “x slaat x” levert op: “Jan slaat Jan”. “Uniforme substitutie van een naam a voor een variabele v” wil dus zeggen dat overal waar v voorkwam a komt te staan.

Bij het behandelen van kwantificerende uitdrukkingen komen de variabelen ons goed van pas. We kunnen de betekenis van “Iedereen slaat Piet” als volgt uitleggen. Welke interpretatie je ook kiest voor x (dat wil zeggen: welke *waarde* je ook aan x toekent), “x slaat Piet” is waar. Net zo voor “Iemand slaat Piet”. Dit is waar wanneer er *een* toekenning bestaat voor x zodanig dat “x slaat Piet” waar is.

We kunnen dit alles nog iets explicieter opschrijven. In plaats van “Iedereen slaat Piet” schrijven we: “Voor alle x geldt: x slaat Piet”. In plaats van “Iemand slaat Piet”: “Voor minstens één x geldt: x slaat Piet”. “Iedereen scheert zichzelf” wordt nu: “Voor alle x geldt: x scheert x”.

We maken een inventaris op van wat we nu hebben:

- *namen* van individuen;
- *variabelen*, die kunnen staan voor willekeurige individuen;
- uitdrukkingen die geïnterpreteerd worden als eigenschappen of relaties (voortaan: *predikaat-uitdrukkingen*);
- “voor alle v geldt: $\cdots$” en “voor minstens één v geldt: $\cdots$”, waarbij v een variabele is; voortaan noemen we deze uitdrukkingen *kwantoren*;
- de waarheidsfunctionele *voegwoorden* nemen we gewoon over uit de propositielogica.

Namen, variabelen, predikaat-uitdrukkingen, kwantoren en voegwoorden, ziedaar de ingrediënten van de predikatenlogica. In de volgende paragraaf gaan we er systematisch op in.

10.2 De syntaxis van de predikatenlogica

De definitie van de welgevormde formules van een predikatenlogische taal gaat weer met *recursie*. De situatie is nu een tikje ingewikkelder dan bij de propositielogica, omdat de allersimpelste formules (de formules die niet uit kleinere formules zijn samengesteld, ofwel: de *atomaire* formules) zelf ook nog interne structuur hebben. In de propositielogica waren de atomaire formules simpelweg propositieletters. In de predikatenlogica zijn het combinaties van namen en/of variabelen met *predikaatletters*. In plaats van “— slaapt” gebruiken we de predikaatletter S. We moeten er dan wel bij vermelden dat S moet worden gecombineerd met één naam of variabele. In jargon: S is een eenplaatsige predikaatletter. Als a een naam is, dan is Sa een atomaire formule. Net zo: als x een variabele is, dan is Sx een atomaire formule.

Nu in het algemeen. Stel, we hebben de beschikking over een verzameling $\{a,b,c,\ldots\}$ van namen (ook wel *individuele constanten* genaamd), een verzameling $\{A,B,\ldots,R,S,\ldots\}$ van predikaatletters, waarbij van elke predikaatletter de zogenaamde *plaatsigheid* (het aantal namen en/of variabelen waarmee hij combineert) (Eng.: *arity*) is gegeven, en tenslotte een verzameling individuele variabelen $\{x,y,z,\ldots\}$. Dan is het recept voor atomaire formules dit.

- Laat P een predikaatletter zijn van plaatsigheid n, en laat $t_1,\ldots,t_n$ variabelen of constanten zijn (met een parapluwoord: *individuele termen*). Dan is $Pt_1\ldots t_n$ een atomaire formule.

Een paar voorbeelden. A is een eenplaatsige predikaatletter, c is een constante. Nu is Ac een atomaire formule (de formule zou kunnen staan voor “cornelis

*A*pplaudiseert"). Zij R een tweeplaatsige predikaatletter, c weer een constante, en x en y variabelen. Dan zijn Rxy, Ryx, Rxx, Ryc, Rcc atomaire formules.

Opgave 10.1 Vertaal in atomaire formules:

1. Jan scheert Piet.
2. Piet scheert Jan.
3. Jan scheert zichzelf.

Gebruik hierbij de volgende **vertaalsleutel**:

S:	scheert [tweeplaatsig]
j:	Jan
p:	Piet

We kunnen nu al een soort propositielogica gaan doen met atomaire predikaatlogische formules, door op de gebruikelijke manier de waarheidsfunctionele voegwoorden in te voeren. Alles gaat dan als in de propositielogica, alleen de propositieletters zijn vervangen door atomaire formules.

Opgave 10.2 We geven weer een vertaalsleutel, maar voor het gemak gebruiken we nu variabelen om de plaatsigheid van de predikaatletters aan te geven:

$Gxyz$:	x geeft y aan z	Mx:	x is een mens
Wx:	x weent	Kx:	x kraait
Ixy:	x is intelligenter dan y	s:	Sokrates
h:	de haan	p:	Petrus.

Vertaal nu:

1. Sokrates is intelligenter dan de haan.
2. Als de haan kraait, dan weent Petrus.
3. Sokrates geeft de haan niet aan Petrus.
4. De haan is geen mens.
5. Als Sokrates een mens is, dan is hij intelligenter dan de haan.
6. Sokrates is intelligenter dan de haan of de haan is intelligenter dan Sokrates.
7. Sokrates geeft zichzelf de haan.

Goed, dit was aardig, maar het wordt pas echt leuk wanneer we kwantoren gaan invoeren. De *universele kwantor* of *al-kwantor*, $\forall$, wordt gecombineerd met een variabele x. $\forall x$ betekent: "Voor alle x in het discussiedomein geldt dat…". We kunnen nu, gegeven de vertaalsleutel uit de laatste opdracht, "Iedereen weent" vertalen als: $\forall xWx$. "Niet iedereen is intelligenter dan Sokrates" wordt: $\neg\forall xIxs$. $\exists$ is de *existentiële kwantor*. $\exists x$ betekent: "Er is minstens één x waarvoor geldt dat…". Dus: "Sokrates geeft iets aan Petrus" wordt: $\exists xGsxp$. "Niemand is intelligenter dan de haan": $\neg\exists xIxh$.

Hoe moeten we nu "Alle mensen wenen" vertalen? We moeten dat in elk geval zo doen dat we er, ongeacht wie of wat Petrus is, het volgende uit kunnen concluderen: "Als Petrus een mens is, dan weent Petrus". De vertaling van deze laatste zin weten we al: $Mp \rightarrow Wp$. Nu is de vertaling van "Alle mensen wenen" niet moeilijk meer: $\forall x(Mx \rightarrow Wx)$. Ga zelf na waarom $\forall x(Mx \wedge Wx)$ *niet* goed is.

Nu de vertaling van "Minstens één mens weent". Even zijn we geneigd om deze zin, naar analogie van het *alle*-geval, als volgt te vertalen: $\exists x(Mx \rightarrow Wx)$. Dit is echter fout, want uit de propositielogica weten we dat deze formule logisch equivalent is met $\exists x(\neg Mx \vee Wx)$, en dit is al waar als er ook maar één niet-menselijk wezen in het discussiedomein te vinden is, terwijl geen mens weent. Wat we juist willen zeggen is dat er menselijke wezens zijn die wenen, dus: $\exists x(Mx \wedge Wx)$.

"Geen mens weent" wordt nu: $\neg\exists x(Mx \wedge Wx)$. "Geen mens weent" betekent hetzelfde als "Voor elk mens geldt dat hij niet weent". We kunnen dus de zin dus evengoed als volgt vertalen: $\forall x\neg(Mx \wedge Wx)$. Dit is weer equivalent met: $\forall x(Mx \rightarrow \neg Wx)$. Al deze vertalingen zijn goed, en wel: precies even goed. Ze komen namelijk op hetzelfde neer. Dit is een verschijnsel dat we al uit de propositielogica kennen. Ook daar heeft het geen zin om te gaan bekvechten over de keuze tussen $p \rightarrow \neg q$ en $\neg(p \wedge q)$, als vertaling van "Als Piet boos is, dan vindt Marietje hem niet aardig".

We kunnen de zaak ook omgekeerd bekijken. Wanneer twee zinnen uit het Nederlands in de predikatenlogica dezelfde vertaling krijgen, dan betekent dat dat de predikatenlogica het eventuele betekenisverschil tussen die twee zinnen verwaarloost. Neem de volgende twee zinnen: "Minstens één mens weent" en "Er is een wenende die mens is". In beide gevallen wordt de vertaling: $\exists x(Mx \wedge Wx)$. Merk op dat je ook over de keuze tussen de vertalingen $\exists x(Mx \wedge Wx)$ en $\exists x(Wx \wedge Mx)$ niet hoeft te bekvechten: die formules zijn namelijk weer equivalent. Je ziet uit het voorbeeld ook dat de predikatenlogica het onderscheid tussen enkelvoud en meervoud verwaarloost: er is geen verschil in vertaling tussen "Een mens weent" en "Sommige mensen wenen". Net zo min tussen "Ieder mens weent" en "Alle mensen wenen"; de vertaling is voor allebei: $\forall x(Mx \rightarrow Wx)$. "Alle wenenden zijn mensen" en "Iedere wenende is een mens" hebben weer elk dezelfde vertaling, maar dit is een andere dan die van "Alle mensen wenen", namelijk: $\forall x(Wx \rightarrow Mx)$. Logisch gezien *moet* er natuurlijk ook verschil in vertaling zijn, want je kunt redeneren:

Alle wenenden zijn mensen.
De haan weent.

De haan is een mens.

maar niet:

Alle mensen wenen.
De haan weent.

De haan is een mens.

Merk op dat er geen verschil in betekenis is tussen aan de ene kant de formule $\forall x(Mx \rightarrow Wx)$ en aan de andere kant $\forall y(My \rightarrow Wy)$. Het feit dat er een andere variabele is gebruikt maakt voor de interpretatie van de formule niets uit.

Tot nu toe hebben we steeds voorbeelden besproken van formules met maar één kwantor, maar er kunnen in een formule willekeurig veel kwantoren voorkomen. Voor de vertaling van "Elke jongen houdt van een meisje" hebben we een universele *en* een existentiële kwantor nodig:

$$\forall x(Jx \rightarrow \exists y(My \wedge Hxy)).$$

De formule drukt uit dat elke jongen een lief heeft (om met de Vlamingen te spreken). Als we willen zeggen dat er een meisje is dat zich in de liefde van alle jongens mag koesteren (bij voorbeeld Brigitte), dan kunnen we dit uitdrukken door een andere formule te gebruiken:

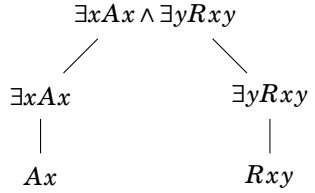
$$\exists x(Mx \wedge \forall y(Jy \rightarrow Hyx)).$$

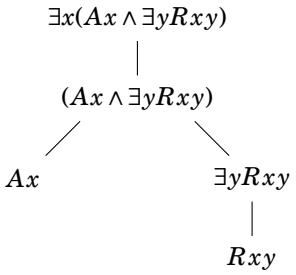
Het voorbeeld laat zien dat de predikatenlogica van pas kan komen om twee betekenissen van de zin "Elke jongen houdt van een meisje" te onderscheiden. Het verschil zit hem in de *bereik*-verhoudingen tussen de kwantoren $\forall$ en $\exists$ die respectievelijk corresponderen met *elke* in *elke jongen* en *een* in *een meisje*. We zeggen dat in de formule $\forall x(Jx \rightarrow \exists y(My \wedge Hxy))$ de universele kwantor groter bereik heeft dan de existentiële, terwijl dat voor de formule $\exists x(Mx \wedge \forall y(Jy \rightarrow Hyx))$ precies andersom ligt. We zeggen ook wel: de universele kwantor heeft in $\forall x(Jx \rightarrow \exists y(My \wedge Hxy))$ groot bereik (Eng.: *wide scope*), de existentiële kwantor heeft in deze formule klein bereik (Eng.: *narrow scope*).

Het *bereik* van een kwantor in een welgevormde formule van de predikatenlogica is, intuïtief gesproken, het gedeelte van die formule waarover de kwantor iets te zeggen heeft, dat wil zeggen het gedeelte waar hij variabelen kan *binden*. Een paar voorbeelden kunnen dit duidelijk maken. We beschouwen de volgende twee formules:

- $\exists xAx \wedge \exists yRxy$
- $\exists x(Ax \wedge \exists yRxy)$.

Het bereik van de kwantoren in deze formules valt af te leiden uit de samenstelling van de formules: een kwantor heeft bereik over de deelformule waarmee hij is samengesteld. Hoewel we de precieze recursieve definitie van predikatenlogische formules nog moeten geven, zijn hier alvast een paar constructiebomen (vergelijk § 7.4 voor constructiebomen in de propositielogica; het begrip *bereik van een kwantor* is geheel analoog aan het begrip *bereik van een zinsoperator*):





Uit deze constructiebomen valt het bereik van $\exists x$ in de twee formules af te leiden. In de eerste boom vormt $\exists x$ samen met de deelformule Ax een formule $\exists xAx$, die vervolgens wordt geconjugeerd met een andere formule. Het bereik van de kwantor $\exists x$ is hier dus de deelformule Ax. In de tweede boom wordt $\exists x$ gecombineerd met $(Ax \wedge \exists yRxy)$ tot een nieuwe formule. Het bereik van de kwantor $\exists x$ is hier dus de formule $(Ax \wedge \exists yRxy)$.

In de twee formules die we zoëven bekeken hebben zouden we het bereik van de kwantor $\exists x$ met behulp van onderstrepingen als volgt kunnen aangeven:

- $\exists x\underline{Ax} \wedge \exists yRxy$
- $\exists x\underline{(Ax \wedge \exists yRxy)}$

Zulke onderstrepingen geven natuurlijk geen nieuwe informatie: omdat constructiebomen voor formules uniek zijn, ligt het bereik van alle kwantoren die in een formule voorkomen ondubbelzinnig vast.

We komen nog even terug op het voorbeeld "Elke jongen houdt van een meisje", met zijn twee predikatenlogische parafrases:

- $\forall x(Jx \rightarrow \exists y(My \wedge Hxy))$
- $\exists y(My \wedge \forall x(Jx \rightarrow Hxy))$

In de eerste parafrase is het bereik van de universele kwantor de deelformule $Jx \rightarrow \exists y(My \wedge Hxy)$, en dat van de existentiële kwantor de deelformule $(My \wedge Hxy)$. We zien dus dat het bereik van de existentiële kwantor een *echt deel* is van het bereik van de universele kwantor. In het geval van $\exists y(My \wedge \forall x(Jx \rightarrow Hxy))$ ligt dit precies andersom: hier is het bereik van de universele kwantor een *echt deel* van het bereik van de existentiële. We kunnen nu een heel precieze definitie geven van wat het betekent dat de ene kwantor groter bereik heeft dan de andere.

Definitie 10.1 *In formule φ heeft (een voorkomen van een) kwantor K_1* **groot bereik** *ten opzichte van (een voorkomen van een) kwantor K_2 (of, wat op hetzelfde neerkomt: heeft K_2* **klein bereik** *ten opzichte van K_1) desda het bereik van K_2 een deelformule is van het bereik van K_1.*

Vrije en gebonden variabelen

Merk op dat in de formule $\exists xAx \wedge \exists yRxy$ de variabele x in Rxy (preciezer gezegd: het *voorkomen* van x in Rxy; Eng.: the *occurrence* of x in Rxy) door geen van beide kwantoren

wordt gebonden. De kwantor $\exists y$ bindt x in Rxy niet, omdat $\exists y$ alleen voorkomens van y binnen zijn bereik bindt; de kwantor $\exists x$ bindt x in Rxy niet, omdat Rxy niet binnen het bereik van deze kwantor valt. Zulke niet-gebonden voorkomens van variabelen heten *vrij* (Eng.: *free*), de andere voorkomens heten *gebonden* (Eng.: *bound*). We zeggen: de variabele x komt vrij voor in $\exists xAx \wedge \exists yRxy$. De variabele x komt overigens ook gebonden voor in $\exists xAx \wedge \exists yRxy$. Dus: dezelfde variabele kan in een formule zowel vrij als gebonden voorkomen. De variabele y komt alleen gebonden voor in $\exists xAx \wedge \exists yRxy$.

We kunnen ook spreken over het vrij/gebonden zijn van voorkomens van variabelen (Eng.: *variable occurrences*) in een *deelformule* van een formule. Voorbeeld: de variabele x komt alleen gebonden voor in de formule $\exists x(Ax \wedge \exists yRxy)$, maar ze is vrij in de deelformule $(Ax \wedge \exists yRxy)$: *binnen die deelformule* komt geen kwantor voor die de beide voorkomens van x bindt.

Opgave 10.3 Geef het bereik aan van de kwantor $\forall x$ in de volgende formules:

1. $\forall xRxx$
2. $\forall x\exists y\forall zSxyz$
3. $\exists z\forall x\forall ySxyz$
4. $Ac \wedge \forall x\exists yRxy$
5. $\exists z\forall x(Ax \wedge Bz)$
6. $\forall x(Ax \rightarrow Bx) \wedge Cx$
7. $\forall x((Ax \rightarrow Bx) \wedge Cx)$
8. $(Ax \rightarrow \forall xBx) \wedge Cx$.

We spreken af dat de combinatie van een kwantor $\forall x$ of $\exists x$ met een formule φ zo moet worden geïnterpreteerd dat de kwantor alle voorkomens van x die *vrij* zijn in φ bindt. Nu zijn er een paar speciale gevallen. Het kan gebeuren dat x helemaal niet voorkomt in φ. Voorbeeld: $\forall x\exists yRyy$. De kwantor $\forall x$ bindt niets, want x komt niet voor in $\exists yRyy$. We spreken in zo'n geval van *loze kwantificatie* (Eng.: *vacuous quantification*). De reden waarom we loze kwantificatie toestaan is een hele goede: het vereenvoudigt de syntactische definitie van de predikatenlogica; je hoeft nu immers het loze geval niet meer uit te sluiten.

Een iets ander speciaal geval bij het combineren van een kwantor $\forall x$ en een formule φ hebben we—je had het waarschijnlijk zelf al bedacht—wanneer x wel voorkomt in φ, maar alleen *gebonden*. Voorbeeld: $\forall x\exists xRxx$. Hier wordt $\forall x$ gecombineerd met $\exists xRxx$, en in die deelformule komt x alleen gebonden voor. Volgens afspraak mag $\forall x$ alleen *vrije* voorkomens van x binden in de deelformule waarmee de kwantor combineert, dus ook hier is de kwantificatie *loos*. Onthoud: een voorkomen van een variabele wordt gebonden door de dichtstbijzijnde kwantor die *loopt* over die variabele en die dat voorkomen van de variabele in zijn bereik heeft. Dus: in de formule $\forall x(Ax \rightarrow (Bx \vee \exists xCx))$ wordt de x in Cx gebonden door de existentiële kwantor. We geven van enige belangrijke begrippen formele definities.

Definitie 10.2 *'φ is een deelformule van ψ' wordt recursief gedefinieerd door (φ en ψ zijn welgevormde formules):*

- *φ is een deelformule van φ.*
- *Als φ een deelformule van ψ is, dan is φ ook een deelformule van $\neg\psi$, $\forall v\psi$, $\exists v\psi$ (v is een variabele).*
- *Als φ een deelformule is van ψ of van χ, dan ook van $(\psi \wedge \chi)$, $(\psi \vee \chi)$, $(\psi \rightarrow \chi)$ of $(\psi \leftrightarrow \chi)$.*

Opgave 10.4 Geef zelf een recursieve definitie van 'v is een vrij voorkomen van een variabele in φ'.

Definitie 10.3 *v is een* **gebonden** *voorkomen van een variabele in φ wanneer v niet vrij is in φ.*

Definitie 10.4 *Formules waarin alle voorkomende variabelen gebonden zijn door een of andere kwantor heten* **gesloten formules** *(Eng.:* closed formulas*), of ook wel:* **zinnen** *(Eng.:* sentences*).*

Definitie 10.5 *Formules waarin een of meer variabelen vrij voorkomen heten* **open formules** *(Eng.:* open formulas*).*

Opgave 10.5 Geef aan welke van de formules uit opdracht 10.3 open zijn en welke gesloten.

Het verschil tussen open en gesloten formules blijkt bij het interpreteren van die formules (zie § 11): het waar of onwaar zijn van gesloten formules hangt niet af van een keuze voor de interpretatie van de variabelen van de taal, het waar of onwaar zijn van open formules hangt daar wel van af. Voor we Ax kunnen interpreteren moeten we weten waar x op slaat, maar bij $\forall xAx$ hoeven we dat niet te weten.

Een volgend punt is: hoe verhouden de existentiële en de universele kwantor zich tot elkaar? Die onderlinge verhoudingen kunnen we uitdrukken met behulp van de negatie-operator. $\neg\exists xAx$ en $\forall x\neg Ax$ zijn logisch equivalent: ontkennen dat er dingen zijn met zekere eigenschap komt immers op hetzelfde neer als zeggen dat van alle dingen geldt dat ze die eigenschap niet hebben. Heel in het algemeen hebben we:

- $\forall v\neg\varphi \iff \neg\exists v\varphi$
- $\exists v\neg\varphi \iff \neg\forall v\varphi$.

Deze twee principes kunnen worden gebruikt om een negatie-teken "door een kwantor heen te trekken". Er volgt ook uit dat we in principe maar een van beide kwantoren nodig hebben. Wanneer we $\forall$ hebben kunnen we $\exists$ met behulp daarvan en van de negatie-operator definiëren, namelijk als $\neg\forall\neg$. Omgekeerd kunnen we ook

$\forall$ definiëren in termen van negatie en de andere kwantor, namelijk als $\neg\exists\neg$.

We kunnen deze principes gebruiken om formules te vervangen door equivalente formules met alleen universele kwantoren, of met alleen existentiële kwantoren. Als voorbeeld schrijven we $\neg\exists x(Ax \wedge Bx)$ met alleen universele kwantoren:

$$\neg\exists x(Ax \wedge Bx) \iff \neg\neg\forall x\neg(Ax \wedge Bx) \iff \forall x\neg(Ax \wedge Bx).$$

In de laatste stap is gebruik gemaakt van de wet van de dubbele negatie.

Opgave 10.6 Geef voor de volgende formules equivalente formules met alleen universele kwantoren:

1. $\forall x\exists yRxy$
2. $\exists xAx \vee \forall yBy$
3. $\exists x\forall y\exists zSxyz$
4. $\neg\exists xAx \vee \forall yBy$
5. $\neg(\exists xAx \vee \forall yBy)$
6. $\exists x\exists y\exists z(Ax \vee Ryz)$.

Opgave 10.7 Geef voor de formules uit de vorige opdracht ook equivalente formules met alleen existentiële kwantoren.

Het wordt hoog tijd voor de precieze definitie van de verzameling welgevormde formules van predikatenlogische talen.

Formele definities

Bij de predikatenlogica hoort een hele familie van verschillende talen, passend bij zeer uiteenlopende, zogenaamde, *structuren* zoals de gehele getallen met optellen en vermenigvuldigingen, of stacks met push en pop. (Hier komen we nog op.) Deze talen heten ook wel 1e-orde talen (1st-order languages), zoals de predikatenlogica soms ook "1e-orde predikatenlogica" genoemd wordt.

Definitie 10.6 *Het alfabet van een 1e-orde taal L bestaat uit de volgende ingrediënten.*

i) *Aftelbaar oneindig veel* ***variabelen****: $x_1, x_2, x_3 \ldots$. De verzameling van variabelen is* VAR*. Variabelen worden ook aangeduid met letters achter in het alfabet: x, y, z, u, v, x', x'', $\ldots$.*

ii) *(Afhankelijk van de taal L) een eindig of aftelbaar oneindig aantal* ***functiesymbolen*** *(ook wel:* ***functieletters****): f, g, h, f_1, f_2, $\ldots$. Elk functiesymbool f bezit een* ariteit *(of* plaatsigheid*); dit is het aantal argumenten waarop f geacht wordt te werken. Functiesymbolen met ariteit* 1*,* 2*,* 3*, n heten respectievelijk* unair*,* binair*,* ternair*, n-*air*. (Of ook: 1-plaatsig, 2-plaatsig, enzovoorts.) De verzameling functiesymbolen is* FUN*.*

Een belangrijk bijzonder geval is dat van een functiesymbool met ariteit nul. Dergelijke functiesymbolen heten constanten *en worden geschreven als a, b, c, d, c_1, c_2, $\ldots$. De verzameling constanten is* $\mathrm{CON}(L)$*. Dus* $\mathrm{CON}(L) \subset \mathrm{FUN}(L)$*.*

In sommige logische teksten worden functiesymbolen afgeteld naar ariteit: eerst de constanten, dan de éénplaatsige symbolen, dan de tweeplaatsige symbolen, etc:

$$\begin{array}{cccc} f_1^0 & f_2^0 & f_3^0 & \cdots \\ f_1^1 & f_2^1 & f_3^1 & \cdots \\ f_1^2 & f_2^2 & f_3^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

(Het blijven er natuurlijk even veel.)

iii) *(Afhankelijk van de taal L) een eindig of aftelbaar oneindig aantal* ***predikaatsymbolen*** *(ook wel:* ***predikaatletters****, relatiesymbolen, of relatieletters): P, Q, R, P_0, P_1, $\ldots$. Net als functiesymbolen bezitten predikaatsymbolen een ariteit. Een bijzonder geval zijn weer de nul-plaatsige predikaatsymbolen of propositiesymbolen; zij zullen dezelfde rol spelen als propositieletters in de propositielogica.*

In sommige logische teksten worden predikaatsymbolen afgeteld naar ariteit:

$$\begin{array}{cccc} P_1^0 & P_2^0 & P_3^0 & \cdots \\ P_1^1 & P_2^1 & P_3^1 & \cdots \\ P_1^2 & P_2^2 & P_3^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

iv) *Een binair predikaatsymbool voor gelijkheid: "$=$".*

v) *De propositionele connectieven: "$\neg$", "$\rightarrow$", "$\wedge$", "$\vee$", "$\equiv$", en de propositieconstante: "$\bot$".*

vi) *Kwantoren (quantifiers); de existentiële kwantor: "$\exists$" ("er is") en de universele kwantor: "$\forall$" ('voor alle').*

vii) *Hulpsymbolen: "$,$", "$($", en "$)$".*

Om geen verwarring te krijgen wordt stilzwijgend aangenomen dat VAR, $\mathrm{FUN}(L)$ en $\mathrm{PRE}(L)$ paarsgewijs disjunct zijn. Als het duidelijk is over welke taal L het gaat, zullen we FUN, PRE, etc. schrijven voor $\mathrm{FUN}(L)$, $\mathrm{PRE}(L)$, etc.

Het gedeelte *i)*, *iv)*, *v)*, *vi)*, *vii)* ligt voor elke 1e-orde taal vast; deze symbolen heten de *logische symbolen*. De overige, de functie- en predikaatsymbolen, zijn dan de *niet-logische symbolen*; deze moeten voor iedere predikaatlogische taal apart worden vastgelegd. (De in *ii)* en *iii)* genoemde predikaat- en functiesymbolen kunnen eventueel ook geheel ontbreken.)

Definitie 10.7 *De verzameling van* ***termen*** *in de taal L, aangeduid met* $\mathrm{TER}(L)$*, is als volgt inductief gedefinieerd:*

i) $\mathrm{VAR} \subseteq \mathrm{TER}(L)$

ii) *Als* $t_1,\ldots,t_n \in \mathrm{TER}(L)$*, en* $f \in \mathrm{FUN}(L)$*, dan is* $f(t_1,...,t_n) \in \mathrm{TER}(L)$.

Merk op dat $\mathrm{CON}(L) \subset \mathrm{TER}(L)$. (Neem $n = 0$ in bovenstaande definitie.)

Termen worden aangegeven met t, t_1, t_2,... , s, s_1, s_2,.... Termen waar geen variabelen voorkomen heten *gesloten termen* of *grondtermen* (closed terms, ground terms).

Definitie 10.8 *De verzameling van* ***formules*** *in de taal L, aangeduid met* FOR(L)*, is als volgt inductief gedefinieerd:*

i) $\bot \in \mathrm{FOR}(L)$.

ii) *Als* $t_1, t_2 \in \mathrm{TER}(L)$*, dan is* $(t_1 = t_2) \in \mathrm{FOR}(L)$.

iii) *Als* $t_1,\ldots,t_n \in \mathrm{TER}(L)$*, en* $P \in \mathrm{PRE}(L)$*, dan is* $P(t_1,...,t_n) \in \mathrm{FOR}(L)$*. In het bijzonder (*$n = 0$*) is een propositiesymbool* $P \in \mathrm{FOR}(L)$.

iv) *Als* $\varphi \in \mathrm{FOR}(L)$*, dan* $(\neg\varphi) \in \mathrm{FOR}(L)$.

v) *Als* $\varphi, \psi \in \mathrm{FOR}(L)$*, dan* $(\varphi \to \psi)$*,* $(\varphi \wedge \psi)$*,* $(\varphi \vee \psi)$*, en* $(\varphi \equiv \psi)$ *in* FOR(L).

vi) *Als* $\varphi \in \mathrm{FOR}(L)$*, en* $x \in \mathrm{VAR}$*, dan* $(\forall x\varphi)$ *en* $(\exists x\varphi) \in \mathrm{FOR}(L)$.

Formules verkregen met clausules *i)-iii)* heten *atomaire formules* (atomen). Buitenste haakjes worden weer weggelaten. Formules worden net als in de propositielogica aangegeven met φ, ψ, α, β, γ, φ_1, φ_2, ..., ψ_1, ψ_2, ... Propositiesymbolen worden hier met hoofdletters aangeduid.

Kwantoren binden sterker dan propositionele voegtekens, zodat we mogen schrijven $\forall x\varphi \to \exists y\psi$ in plaats van $(\forall x\varphi) \to (\exists y\psi)$. De notatieconventies die we in de propositielogica hanteren (zoals het achterwege laten van haakjes als de prioriteit duidelijk is) gelden ook hier. Verder schrijven we Rxy in plaats van $R(x,y)$ en $forallx\forall y\varphi$ of zelfs $\forall xy\varphi$ in plaats van $\forall x(\forall y\varphi)$. Analoog voor de existentiële kwantor.

Merk op dat er in de bovenstaande definitie weer metavariabelen zijn gebruikt: P is een metavariabele over predikaatletters, t_1 tot en met t_n zijn metavariabelen over individuele termen (constanten of variabelen), v is een metavariabele over variabelen, en φ en ψ zijn metavariabelen over formules. Zie verder Tabel 10.1 op blz. 122.

Het is bijzonder nuttig nog wat verder te oefenen in het vertalen van het Nederlands naar de predikatenlogica. Dergelijke vertalingen leggen logische overeenkomsten bloot tussen zinnen die er 'oppervlakkig' verschillend uitzien:

Op elk potje past een dekseltje. (1)

Ieder huisje heeft zijn kruisje. (2)

Alle gebeurtenissen hebben een oorzaak. (3)

Iedereen houdt van iemand.

Deze zinnen vertonen overeenkomst in *logische vorm*, hetgeen blijkt uit het feit dat ze allemaal kunnen worden vertaald met behulp van de formule

$$\forall x(Px \to \exists y(Qy \wedge Rxy))$$

waarbij P, Q en R dan natuurlijk steeds andere predikaten uitdrukken.

In het algemeen moet je, om van het Nederlands naar de predikatenlogica te kunnen vertalen, ten eerste aangeven wat het *discussiedomein* is, en ten tweede hoe de predikaatletters en constanten die je in je vertaling gebruikt, moeten worden gelezen (dat wil zeggen: je moet een *vertaalsleutel* geven).

Stel dat we willen vertalen: *Al het aardse is vergankelijk*. We kiezen als discussiedomein de verzameling van aardse en bovenaardse dingen. De vertaalsleutel wordt: Ax: x is aards; Vx: x is vergankelijk. Op grond hiervan wordt de vertaling: $\forall x(Ax \to Vx)$.

Merk op dat de keuze van het discussiedomein de vertaling kan beïnvloeden. Als bij de vertaling van "Al het aardse is vergankelijk" de verzameling van aardse dingen als discussiedomein wordt genomen, dan wordt de vertaling simpelweg: $\forall xVx$.

Verder kun je natuurlijk sjoemelen bij het vertalen door het kiezen van een vertaalsleutel waarin een deel van de logische structuur van een zin is weggemoffeld. Stel bij voorbeeld dat je moet vertalen: "Niemand kijkt een (hem) gegeven paard in de bek". Je kun je hier vanaf maken door de verzameling van mensen als discussiedomein te kiezen, en als vertaalsleutel te nemen:

– Kx: x kijkt een hem gegeven paard in de bek.

De vertaling wordt nu simpelweg: $\neg\exists xKx$. Helaas, eenvoud is hier niet het kenmerk van het ware, want het kiezen van de bovenstaande vertaalsleutel betekent dat de kans verkeken is om de geldigheid van de volgende redenering predikatenlogisch te verantwoorden:

Niemand kijkt een (hem) gegeven paard in de bek.
Elias geeft Blessie aan Jan.
Blessie is een paard.

Jan kijkt Blessie niet in de bek.

Nee, we moeten juist in de predikatenlogische vertaling zo weinig mogelijk van de logische structuur verdonkeremanen. Een veel betere vertaling van "Niemand kijkt een (hem) gegeven paard in de bek" krijgen we wanneer we als discussiedomein nemen: de verzameling van alle mensen en paarden, en als vertaalsleutel:

– Mx: x is een mens;

– $Gxyz$: x geeft y aan z;

– Px: x is een paard;

– Kxy: x kijkt y in de bek.

		Formeel afgeteld	Informeel
VAR	alle **variabelen**	$\{x_1, x_2, \ldots\}$	$x, y, z, u, v, x', x'', \ldots$
CON(L)	alle **constanten** in L	$\{c_1, c_2, \ldots\}$, soms $\{f_1^0, f_2^0, \ldots\}$	$a, b, c, d, c_1, c_2, \ldots$
FUN(L)	alle **functiesymbolen** in L	$\{f_1, f_2, \ldots\}$, soms $\{f_1^0, f_2^0, \ldots, f_1^1, f_2^1, \ldots\}$	$f, g, h, g_1, g_2, \ldots$
PRE(L)	alle **predikaatletters** in L	$\{P_1, P_2, \ldots\}$, soms $\{P_1^0, P_2^0, \ldots, P_1^1, P_2^1, \ldots\}$	$P, Q, R, P_0, P_1, P_2, P_3, \ldots$
TER(L)	alle **termen** in L	n.v.t.	$t, t_1, t_2, \ldots, s, s_1, s_2, \ldots$
FOR(L)	alle **formules** in L	n.v.t.	$\varphi, \psi, \alpha, \beta, \gamma, \varphi_1, \varphi_2, \ldots, \psi_1, \psi_2, \ldots$
SEN(L)	alle **gesloten formules** in L	n.v.t.	id.

Tabel 10.1: De taal van de predikatenlogica L.

De vertaling wordt nu:

$$\forall x \forall y \forall z((Mx \wedge Py \wedge Mz \wedge Gxyz) \rightarrow \neg Kzy).$$

Letterlijk: voor alle drietallen bestaande uit een gever, een gegeven paard, en een ontvanger geldt dat de ontvanger het hem gegeven paard niet in de bek kijkt.

Opgave 10.8 Vertaal de volgende zinnen uit het Nederlands naar de taal van de predikatenlogica. Geef steeds het discussiedomein en de vertaalsleutel aan.

1. Bhagwan is een profeet of hij is een profiteur.
2. Als Bhagwan een profeet is, dan adoreert Annie hem.
3. Iedereen die Bhagwan volgt adoreert hem.
4. Wie Bhagwan niet adoreert, volgt hem niet.
5. Alles wat Bhagwan zegt spreekt Jan tegen.
6. Als Bhagwan iets zegt, dan spreekt Jan het tegen.
7. Geen van zijn volgelingen spreekt tegen wat Bhagwan zegt.

Je kunt bij het maken van deze opdracht gebruik maken van de aanwijzing die hieronder wordt gegeven.

Aanwijzing: Wanneer je opdracht 10.8 probeert te maken zul je merken dat het vinden van goede predikatenlogische vertalingen soms enige inventiviteit vergt. In onderdeel 4. moet je bij voorbeeld bedenken dat *wie* staat voor *al wie*, zodat er in feite een universele bewering wordt gedaan. In 6. word je geconfronteerd met het probleem dat het voornaamwoord *het* in de nazin het woord *iets* in de voorzin als antecedent moet hebben: in de predikatenlogica moet dit verband worden aangegeven door ervoor te zorgen dat de kwantor die de vertaling vormt van het antecedent de variabele bindt die de vertaling vormt van het voornaamwoord. Bij rechttoe, rechtaan vertalen lukt dit niet, want dan is het bereik van de kwantor te klein. Je komt hier alleen uit wanneer je 'naar de geest vertaalt', door eerst over te stappen naar een equivalente zin waarin dit probleem niet optreedt.

Opgave 10.9 Nog een paar vertaalklussen. Geef weer discussiedomein en vertaalsleutel aan.

1. Alle barbiers die zichzelf niet scheren, worden door een barbier geschoren.
2. Alle barbiers die zichzelf niet scheren, worden door geen enkele barbier geschoren.
3. Geen barbier die zichzelf niet scheert wordt door alle barbiers geschoren.
4. Als een barbier zichzelf niet scheert, dan scheert hij niet iedereen.
5. Als een barbier zichzelf niet scheert, dan scheert hij niet iedereen die zichzelf niet scheert.

Opgave 10.10 Vertaal de volgende Nederlandse spreekwoorden in de taal van de predikatenlogica. Geef steeds het discussiedomein en de vertaalsleutel aan. Wellicht ten overvloede: je moet de letterlijke zin van de spreekwoorden weergeven. Dit neemt echter niet weg dat je moet weten wat een spreekwoord *betekent* voor je aan het vertalen slaat, want de betekenissen bepalen hoe eventuele syntactische dubbelzinnigheden moeten worden opgelost.

1. Wie zijn kinderen liefheeft, kastijdt hen.

2. Geen rozen zonder doornen.

3. Alle hout is geen timmerhout.

4. Het is niet alles goud wat blinkt.

5. Kinderen die vragen worden overgeslagen.

6. Het zijn niet allen koks, die lange messen dragen.

De predikatenlogica heeft een zeer grote kracht. Wanneer het je niet gelukt is om alle opgedragen vertalingen te maken ligt dat niet aan de predikatenlogica maar aan jou .... De benodigde inventiviteit voor het vinden van adequate predikatenlogische vertalingen kan echter worden aangeleerd.

De kracht van de predikatenlogica blijkt bij voorbeeld uit het feit dat de hele verzamelingenleer in predikatenlogica valt uit te drukken.

Opgave 10.11 Verzamelingtheoretisch jargon kan worden omgezet in predikatenlogica. Bij voorbeeld: $a \in A$ wordt predikatenlogisch: Aa. Vertaal met behulp van dit gegeven:

1. $A \subseteq B$

2. $A = B$

3. Als $A \subseteq B$ en $B \subseteq C$ dan $A \subseteq C$.

Complexe predikaten kunnen soms worden opgebouwd uit eenvoudiger predikaten plus wat logisch bindmiddel. Zo kun je *grootvader* definiëren in termen van *ouder* en *mannelijk*. Laat de volgende vertaalsleutel gegeven zijn:

– Mx: x is van het mannelijk geslacht;

– Oxy: x is ouder van y.

Aangenomen dat de verzameling mensen het discussiedomein is levert dit de volgende vertaling op voor "x is grootvader van y": $\exists z(Oxz \wedge Ozy \wedge Mx)$.

Opgave 10.12 Neem als discussiedomein de verzameling van alle mensen, en gebruik de vertaalsleutel die hierboven gegeven is. Vertaal nu:

1. Niemand is ouder van zichzelf.

2. Iedereen heeft een vader.

3. Iedereen heeft een moeder.

4. Iemand heeft een kind.

5. Niet iedereen is vader van iemand.

6. Niet iedereen heeft een dochter en een zoon.

7. Iedereen heeft een grootmoeder.

8. Niet iedereen heeft een kleinkind.

Opgave 10.13 Laat de volgende vertaalsleutel gegeven zijn:

– Kxy : x is kind van y;

– m: Marie;

– Mx: x is van het mannelijk geslacht;

– p: Piet.

Het discussiedomein is: alle mensen. Parafraseer nu de volgende predikatenlogische formules in het Nederlands:

1. $\exists xKxm$

2. $\exists x(Kxm \wedge Kpx)$

3. $\exists x((Kxm \wedge Mx) \wedge Kpx)$

4. $\exists x((Kxm \wedge Kxp) \wedge \neg Mx)$

5. $\exists x(Kxm \wedge \forall y(Kym \rightarrow Kyp))$

6. $\exists x((Kxm \wedge Mx) \wedge \neg \exists y(Kxy \wedge My))$.

Opgave 10.14 Geef de volgende redeneringen predikatenlogisch weer (vertaalsleutels en discussiedomeinen vermelden):

1. Alle mensen zijn sterfelijk.
   Sokrates is een mens.

   Sokrates is sterfelijk.

2. Geen vis is een zoogdier.
   Alle knaagdieren zijn zoogdieren.

   Geen knaagdier is een vis.

We kunnen nu redeneringen uit het Nederlands vertalen naar formules uit de predikatenlogica. Wat we vervolgens graag zouden willen is: nagaan of die predikaatlogische redeneringen *geldig* zijn. De vraag naar geldigheid is een semantische vraag; zij komt in de nu volgende paragraaf aan de orde.

Hoofdstuk 11

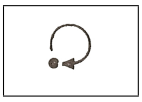
De semantiek van de predikatenlogica

De semantiek van de predikatenlogica is voor het eerst expliciet—dat wil zeggen: in verzamelingstheoretische termen— in 1933 geformuleerd door de Poolse logicus Alfred Tarski (1901-1983).

Alfred Tarski (foto 1918).

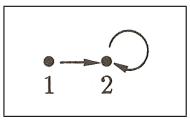
Tarski werkte het idee uit om predikatenlogische formules te interpreteren in verzamelingstheoretische structuren, de zogenaamde *modellen* voor de predikatenlogica.

Vooraleer we in verzamelingstheoretische termen gaan uitleggen wat een model voor een predikatenlogische taal $\mathscr{T}$ is, beginnen we met het tekenen van *plaatjes* voor modellen, om zo een intuïtief idee te krijgen van de predikatenlogische semantiek. Bekijk het volgende plaatje:



Laten we zeggen dat dit een plaatje is van een situatie waarin er maar één ding bestaat (dus: het discussiedomein bestaat uit een enkel object), en dat ding heeft een bepaalde relatie (in het plaatje weergegeven met een pijl) tot zichzelf. Wanneer we ons nu voorstellen dat de predikaatletter R verwijst naar de relatie die in het plaatje door de pijl wordt weergegeven, dan kunnen we met behulp van predikatenlogische formules *uitspraken* gaan doen over de situatie in het plaatje. De volgende formules zijn bijvoorbeeld *waar* in de situatie van het plaatje: $\forall xRxx$; $\exists xRxx$; $\forall x\exists yRxy$; $\exists x\forall yRxy$ (ga dit na). De volgende formule is *niet waar* in de situatie van het plaatje: $\exists x\neg Rxx$. Uiteraard zijn ook alle ontkenningen van de formules die waar zijn in de situatie van het plaatje, onwaar.

In plaatjes met meerdere objecten geven we voor het gemak de objecten aan met cijfers:

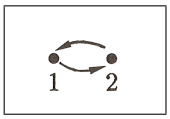


Het plaatje geeft de situatie aan waarin er twee objecten 1 en 2 zijn, waarbij 1 niet in de pijlrelatie staat tot zichzelf, maar wel tot 2, terwijl 2 in de pijlrelatie staat tot zichzelf, maar niet tot 1.

Opgave 11.1 Wanneer we R weer interpreteren als de pijlrelatie, welke van de volgende predikatenlogische formules zijn dan waar in de hierboven gegeven situatie?

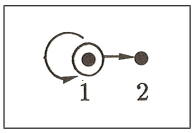
1. $\forall xRxx$.
2. $\exists xRxx$.
3. $\forall x\exists yRxy$.
4. $\exists x\forall yRxy$.
5. $\forall x(\exists yRxy \rightarrow Rxx)$.
6. $\exists x\neg Rxx$.
7. $\forall x\forall y(Rxy \rightarrow Rxx)$.
8. $\forall x(Rxx \rightarrow \exists yRyx)$.

Opgave 11.2 Beschouw nu het volgende plaatje:



Geef aan welke van de formules uit de vorige opdracht waar zijn in de situatie van dit plaatje.

Laten we, behalve de pijlrelatie, ook nog de cirkeleigenschap invoeren. Alle objecten die omcirkeld zijn hebben de cirkeleigenschap. Een mogelijk plaatje is nu:

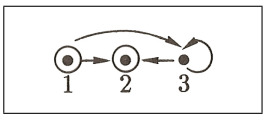


Object 1 heeft de pijlrelatie tot zichzelf en tot 2; 1 heeft bovendien de cirkeleigenschap. 2 heeft de cirkeleigenschap niet, en staat ook tot niets in de pijlrelatie. Veronderstel nu dat de predikaatletter P wordt geïnterpreteerd als de cirkeleigenschap. Dan kunnen we nu onderzoeken of predikatenlogische formules waarin de predikaatletters P en R voorkomen waar zijn in de situatie van dit laatste plaatje.

Opgave 11.3 Ga na welke van de volgende formules waar zijn in het laatst getekende plaatje:

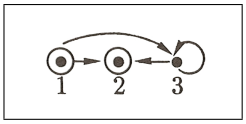
1. $\exists x P x$
2. $\forall x P x$
3. $\forall x(Rxx \rightarrow Px)$
4. $\forall x(Px \rightarrow Rxx)$
5. $\exists x(Px \wedge \neg Rxx)$
6. $\forall x(Px \rightarrow \exists y Rxy)$
7. $\forall x(\exists y Ryx \rightarrow Px)$
8. $\forall x(Rxx \leftrightarrow \neg Px)$
9. $\exists x \exists y(Rxy \wedge \neg Px \wedge \neg Py)$
10. $\forall x(Rxx \rightarrow \exists y(Rxy \wedge Py))$
11. $\forall x(Px \rightarrow \exists y Ryx)$
12. $\exists x \exists y(Rxy \wedge \neg Ryx \wedge \exists z(Rxz \wedge Rzy))$.

Opgave 11.4 Als de vorige opdracht maar nu voor het volgende plaatje:



De predikatenlogische taal die we nodig hebben om de bovenstaande plaatjes te beschrijven is een hele simpele: hij bevat een eenplaatsige predikaatletter P en een tweeplaatsige predikaatletter R. Laten we nu aannemen dat onze taal ook nog constanten a, b en c heeft. Om te weten hoe we de beschrijving van de plaatjes met behulp van deze taal moeten opvatten, moeten we twee dingen weten. Ten eerste: welke objecten in het plaatje worden door welke constanten benoemd? We kunnen bijvoorbeeld afspreken dat in het plaatje uit de laatste opdracht a een naam is voor object 1, b een naam voor object 2, en c een naam voor object 3. Ten tweede: op welke eigenschap van objecten in het plaatje slaat de predikaatletter P (we hebben hierboven afgesproken: op de cirkeleigenschap), en op welke relatie tussen objecten slaat de predikaatletter R (we hebben hierboven afgesproken: op de pijlrelatie)?

Nu algemener: een paar dat bestaat uit een verzameling dingen D en een functie I die aan de constanten van een predikatenlogische taal $\mathscr{T}$ objecten uit D toekent, aan de eenplaatsige predikaatletters van $\mathscr{T}$ deelverzamelingen van D, aan de tweeplaatsige predikaatletters van $\mathscr{T}$ deelverzamelingen van D^2 (dat wil zeggen: tweeplaatsige relaties op D), aan de drieplaatsige predikaatletters van $\mathscr{T}$ deelverzamelingen van D^3 (drieplaatsige relaties op D), enzovoort, noemen we een *model* voor de taal $\mathscr{T}$. Als $\mathscr{M} = \langle D, I\rangle$ een model is, dan heten D en I respectievelijk het *domein* en de *interpretatiefunctie* van het model. Vaak wordt een model voor een predikatenlogische taal een *structuur* genoemd. We zien nu dat het volgende plaatje, gegeven onze afspraken over wat de constanten a, b, c en de predikaatletters P en R betekenen, een model vertegenwoordigt voor de taal die alleen a, b en c als constanten heeft, en alleen P en R als predikaatletters:



Immers, dat model $\mathscr{M} = \langle D, I\rangle$ ziet er als volgt uit. Het domein bestaat uit de objecten 1, 2 en 3, dus:

$$D = \{1,2,3\}.$$

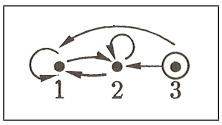
De interpretatiefunctie I heeft als domein de verzameling

$$\{a,b,c,P,R\},$$

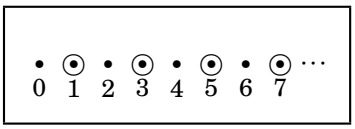
en kent de volgende waarden toe:

- $I(a) = 1$ [Constante a benoemt object 1.]
- $I(b) = 2$ [Constante b benoemt object 2.]
- $I(c) = 3$ [Constante c benoemt object 3.]
- $I(P) = \{1,2\}$ [Predikaatletter P wordt geïnterpreteerd als de verzameling van alle objecten die omcirkeld zijn.]
- $I(R) = \{\langle 1,2\rangle, \langle 1,3\rangle, \langle 3,3\rangle, \langle 3,2\rangle\}$ [Predikaatletter R wordt geïnterpreteerd als de verzameling van alle paren waarvan het eerste lid in de pijlrelatie staat tot het tweede.]

Opgave 11.5 Omschrijf de structuur die bepaald wordt door het volgende plaatje (neem aan dat de afspraken over wat de constanten a, b en c en de predikaatletters P en R betekenen hetzelfde blijven):



In alle plaatjes die we tot nu toe getekend hebben kwamen slechts eindig veel objecten voor. Dat was alleen voor het gemak. Een structuur voor een predikatenlogische taal kan heel goed een oneindig domein hebben. Hier is een voorbeeld:



Dit plaatje symboliseert de verzameling $\mathbb{N} = \{0,1,2,3,4,\dots\}$ van de natuurlijke getallen. De *oneven* getallen zijn omcirkeld. Spreek af dat we de predikaatletter P interpreteren als de eigenschap *oneven zijn*. Spreek verder af dat we R interpreteren als de relatie *groter zijn dan*. Laat opnieuw a, b en c respectievelijk de objecten 1, 2 en 3 benoemen. Dan bepaalt het plaatje (met de afspraken) het volgende model voor de taal $\mathscr{T}$: $\mathscr{M} = \langle D, I\rangle$, waarbij $D = \{0,1,2,3,\dots\}$, $I(a) = 1$, $I(b) = 2$, $I(c) = 3$, $I(P) = \{1,3,5,\dots\}$, $I(R) = \{\langle 1,0\rangle, \langle 2,0\rangle, \langle 2,1\rangle, \langle 3,0\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 4,0\rangle, \dots\}$. Merk op dat in deze structuur niet alle objecten een naam hebben. Alleen 1, 2 en 3 worden benoemd, de rest is naamloos.

Opgave 11.6 Ga na welke van de volgende formules waar zijn in de hierboven gegeven structuur:

1. Pa.
2. Pb.
3. $Pb \rightarrow \forall x Px$.
4. $\forall x \neg Rxx$.
5. $\forall x(Rxx \rightarrow Px)$.
6. $\forall x \forall y(Rxy \vee Ryx)$.
7. $\exists x \forall y Rxy$.
8. $\forall x \exists y Ryx$.
9. $\forall x \exists y Rxy$.
10. $\forall x \exists y(Py \wedge Rxy)$.
11. $\exists x \forall y(\neg Py \rightarrow Ryx)$.
12. $\forall x \forall y \forall z((Rxy \wedge Ryz) \rightarrow Rxz)$.

De oneindigheid van het laatste model is van het simpelste soort. de verzameling natuurlijke getallen is immers *aftelbaar* (zie hoofdstuk 3). De domeinen kunnen natuurlijk ook *overaftelbaar* zijn. Een beroemde stelling (van Löwenheim en Skolem) zegt dat wanneer er een *overaftelbaar* model voor een verzameling formules bestaat, er ook een *aftelbaar* model moet zijn. We zullen er hier niet dieper op ingaan. Wel zullen we verderop zien dat we niet altijd met een *eindig* model kunnen volstaan.

Opgave 11.7 Verzamelingen-jargon predikatenlogisch weergeven kan op de manier van opdracht 10.11, maar het kan ook anders. Beschouw een model $\mathcal{M}$ met verzamelingen als individuendomein, en met een twee-plaatsige relatie $\in$ als gegeven. Interpreteer de tweeplaatsige predikaatletter R als $\in$. Vertaal nu de volgende formules terug in verzamelingenjargon.

1. $\exists x \forall y \neg Ryx$.
2. $\neg \exists x Rxx$.
3. $\forall x \forall y(Rxy \rightarrow \neg Ryx)$.
4. $\forall x \forall y(Rxy \rightarrow \exists z Ryz)$.

11.1 Waarheidswaarde

We hebben hierboven gedefiniëerd wat we bedoelen met: "$\mathcal{M}$ is een model voor predikatenlogische taal $\mathcal{T}$". In de praktijk van de opdrachten hierboven zijn we ook al geconfronteerd met de vraag: wat betekent het dat een formule φ van taal $\mathcal{T}$ waar is in een model $\mathcal{M}$ voor $\mathcal{T}$? Wat we nu nog moeten doen is de achterliggende theorie onder woorden brengen; we moeten een *definitie* geven van *waarheid* voor willekeurige formules van $\mathcal{T}$ in modellen $\mathcal{M}$ voor $\mathcal{T}$. Preciezer gezegd: we moeten de interpretatiefunctie I van een model $\mathcal{M}$ gaan gebruiken voor het definiëren van een *valuatiefunctie* $V_{\mathcal{M}}$ die aan elke formule van onze taal een waarheidswaarde toekent.

Een kleine complicatie hierbij is dat we een afspraak moeten maken omtrent de interpretatie van de vrije variabelen in de open formules van de taal $\mathcal{T}$. Je zou kunnen denken dat dit probleem kan worden omzeild door $V_{\mathcal{M}}$ alleen te definiëren voor de gesloten formules van $\mathcal{T}$, maar dat wil niet: gesloten formules waarin variabelen voorkomen zijn namelijk samengesteld uit kwantoren plus *open* formules, en de waarheidswaarde van de gesloten formule hangt dan af van de—minder complexe—open formule.

Bedeling

De oplossing is als volgt. Neem aan dat we formules van $\mathcal{T}$ evalueren tegen de achtergrond van een zogenaamde *bedeling* of *toekenning* (Eng.: *assignment*) voor de variabelen uit de taal $\mathcal{T}$. Een bedeling is een functie

$$b : \mathrm{VAR} \rightarrow D$$

die de variabelen afbeeldt op objecten in het domein van het model. We kunnen zo'n bedeling b aangeven door—gegeven een volgorde van de variabelen—de objecten te noemen waar de variabelen stuk voor stuk aan worden gekoppeld. Dus bijvoorbeeld:

$$b = \langle d_1, d_3, d_5, d_3, d_{101}, d_2, \ldots \rangle$$

beeldt x_1 af op d_1, x_2 op d_3, etc., waarbij $d_1, d_2, d_3, \ldots$ objecten uit D zijn. In dit geval blijven de waarden van b voor x_i, $i \geq 7$ ongespecificeerd. Merk op dat een object uit D aan meerdere variabelen kan worden toebedeeld.

Omdat de valuatie $V_{\mathcal{M}}$ die we willen gaan definiëren niet alleen van $\mathcal{M}$ maar ook van een bedeling b afhangt, stappen we over van de notatie $V_{\mathcal{M}}$ naar $V_{\mathcal{M},b}$. De functie $V_{\mathcal{M},b}$ die we willen gaan definiëren moet formules gaan afbeelden op waarheidswaarden:

$$V_{\mathcal{M},b} : \mathrm{FOR} \rightarrow \{\mathrm{true}, \mathrm{false}\}.$$

Het definiëren van de functie $V_{\mathcal{M},b}$ is niets anders dan: het geven van een *waarheidsdefinitie* voor de predikatenlogica. Deze *waarheidsdefinitie* voor predikatenlogische formules is van Alfred Tarski.

Eerst moeten we gaan kijken hoe de termen (constanten en variabelen) worden geïnterpreteerd. Voor de termen hebben we een functie nodig die objecten uit D als waarden oplevert:

$$W_{\mathcal{M},b} : \mathrm{TER} \rightarrow D.$$

Dit heet: een waardetoekenning aan de termen van de taal. De definitie is als volgt. Als t een constante is, dan nemen we gewoon het object dat I aan die constante toekent. Dus:

– $W_{\mathcal{M},b}(t) = I(t)$, als t een constante is.

In het andere geval, dus als t een variabele is, hebben we de bedeling b nodig. Immers, I kent aan de variabelen geen objecten toe; de variabelen hadden juist geen vaste interpretatie. Dus:

– $W_{\mathcal{M},b}(t) = b(t)$, als t een variabele is.

Als t een samengestelde term is, dus als

$$t = f(t_1, \ldots, t_n)$$

voor één of ander functiesymbool f en voor één of ander stel termen $t_1, \ldots, t_n$, dan definiëren we $W_{\mathcal{M},b}$ natuurlijk recursief (i.e., naar de opbouw van de term t):

– $W_{\mathcal{M},b}(f(t_1,\ldots,t_n)) = I(f)(W_{\mathcal{M},b}(t_1),\ldots,W_{\mathcal{M},b}(t_n))$.

Dus de waarde van de (symbolische!) term $f(t_1,\ldots,t_n)$ in model $\mathcal{M}$ met variabele-bedeling b is gelijk aan de toepassing van de geïnterpreteerde (en echte!) functie $I(f)$ op de waarden

$$W_{\mathcal{M},b}(t_1),\ldots,W_{\mathcal{M},b}(t_n).$$

Met behulp van $W_{\mathcal{M},b}$ en I gaan we nu $V_{\mathcal{M},b}$ definiëren. Dat gaat weer recursief, net als de definitie van de valuatiefunctie in de propositielogica.

Eerst het atomaire geval. We willen dat een formule zoals Pa waar is in een model $\mathcal{M}$ precies wanneer het object in $\mathcal{M}$ dat a wordt genoemd in de interpretatie van P zit. En voor Px: dit moet waar zijn in $\mathcal{M}$, gegeven een bedeling b, wanneer het object dat door de bedeling aan x wordt toegekend, een element is van de interpretatie van P.

Nu algemeen. De algemene vorm van een atomaire formule is $At_1\ldots t_n$, dus:

$$V_{\mathcal{M},b}(At_1\ldots t_n) = 1 \Leftrightarrow \langle W_{\mathcal{M},b}(t_1),\ldots,W_{\mathcal{M},b}(t_n)\rangle \in I(A)$$

Dan voor de niet-atomaire formules. De waarheidsfunctionele voegwoorden leveren geen problemen op. Die gaan precies zoals we op grond van het propositielogische geval zouden verwachten:

$$\begin{aligned}
V_{\mathcal{M},b}(\neg\varphi) = 1 &\Leftrightarrow V_{\mathcal{M},b}(\varphi) = 0\\
V_{\mathcal{M},b}((\varphi\wedge\psi)) = 1 &\Leftrightarrow V_{\mathcal{M},b}(\varphi) = 1 \text{ en } V_{\mathcal{M},b}(\psi) = 1\\
V_{\mathcal{M},b}((\varphi\vee\psi)) = 1 &\Leftrightarrow V_{\mathcal{M},b}(\varphi) = 1 \text{ of } V_{\mathcal{M},b}(\psi) = 1\\
V_{\mathcal{M},b}((\varphi\rightarrow\psi)) = 1 &\Leftrightarrow V_{\mathcal{M},b}(\varphi) = 1 \Rightarrow V_{\mathcal{M},b}(\psi) = 1\\
V_{\mathcal{M},b}((\varphi\leftrightarrow\psi)) = 1 &\Leftrightarrow V_{\mathcal{M},b}(\varphi) = V_{\mathcal{M},b}(\psi)
\end{aligned}$$

Nu de kwantoren nog. Eerst een simpel voorbeeld. Stel, we willen uitleggen wanneer $\exists xPx$ waar is. We willen daarbij gebruik maken van wat we weten over de waarheid van Px. We kunnen *niet* simpelweg zeggen: $V_{\mathcal{M},b}(\exists xPx) = 1$ desda $V_{\mathcal{M},b}(Px) = 1$. Immers, $V_{\mathcal{M},b}(Px)$ hangt af van de waarde die b toekent aan x: $V_{\mathcal{M},b}(Px) = 1$ desda $b(x) \in I(P)$. Nu zou het echter kunnen zijn dat weliswaar $b(x)$ de eigenschap $I(P)$ niet heeft, maar een *ander* object uit D wel. Dus: we moeten niet alleen kijken naar wat bedeling b met x doet, maar we moeten ook *andere* bedelingen in de beschouwingen betrekken.

Met name zijn we geïnteresseerd in bedelingen die van b alleen verschillen wat betreft de waarde die ze aan x toekennen. Het is handig om hiervoor een notatie in te voeren. We spreken daarom af:

Definitie 11.1 (Variant van een bedeling) *$b(x|d)$ staat voor de bedeling die precies is als b, behalve dat de waarde voor het argument x het object d is (waar b mogelijkerwijs een andere waarde toekende). Preciezer:*

$$b(x|d)(y) = \begin{cases} b(y) & \text{als } y \neq x,\\ d & \text{als } y = x. \end{cases}$$

Gewapend met de nieuwe notatie gaan we nu de kwantoren te lijf.

$$\begin{aligned}
V_{\mathcal{M},b}(\exists v\varphi) = 1 &\\
&\Leftrightarrow \text{voor tenminste één } d \in D : V_{\mathcal{M},b(v|d)}(\varphi) = 1\\
V_{\mathcal{M},b}(\forall v\varphi) = 1 &\Leftrightarrow \text{voor alle } d \in D : V_{\mathcal{M},b(v|d)}(\varphi) = 1
\end{aligned}$$

Hiermee hebben we alle mogelijkheden gehad, en de waarheidsdefinitie voor predikatenlogische formules is compleet. Merk op dat—net als bij de propositielogica—de recursieve waarheidsdefinitie de recursieve definitie van de verzameling welgevormde formules op de voet volgt.

11.2 Vervulbaarheid

We gaan vervulbaarheid definiëren en het daarmee verband houdende semantisch geldigheidsbegrip $\models$. Net als in de propositielogica wordt de dubbele turnstile gebruikt om *logische waarheid* en *logisch gevolg* weer te geven. Bij open formules moeten we rekening houden met bedelingen. Dus

$$\models \varphi.$$

betekent: φ is *logisch waar* (of: *universeel geldig*), dat wil zeggen: φ is waar in alle modellen van de taal, voor alle mogelijke bedelingen. De uitdrukking

$$\varphi \models \psi$$

betekent: ψ volgt logisch uit φ, dat wil zeggen voor alle modellen en bedelingen die φ waar maken, is ψ ook waar.

Hier zijn enkele voorbeelden van universeel geldige formules:

- $\forall x(Px \vee \neg Px)$
- $\forall xPx \leftrightarrow \neg\exists x\neg Px$
- $\forall x(Px \wedge Qx) \rightarrow \forall xPx$
- $Pa \rightarrow \exists xPx$.

Vanwege het onderscheid tussen open en gesloten formules hebben we nog wat jargon nodig.

Definitie 11.2 (Vervulbaarheid) – *Als een paar $\mathcal{M},b$ bestaat, zó dat $V_{\mathcal{M},b}(\varphi) = 1$ dan zeggen we dat het paar $\mathcal{M},b$ de formule φ* vervult*. In dat geval schrijven we*

$$\mathcal{M},b \models \varphi.$$

In dat geval zeggen we ook dat φ vervulbaar *is.*

– *We zeggen dat het paar $\mathcal{M},b$ de formuleverzameling Γ* vervult *als het alle elementen in die verzameling vervult. In dat geval schrijven we*

$$\mathcal{M},b \models \Gamma,$$

en zeggen we dat Γ vervulbaar *is.*

Opgave 11.8 Bewijs dat de lege verzameling vervulbaar is.

Definitie 11.3 (Waar, onwaar, geldig, contingent)

– *Een formule φ heet* **waar** *(Eng.:* **true***) met betrekking tot $\mathcal{M}$, notatie*

$$\mathcal{M} \models \varphi,$$

als $\mathcal{M}, b \models \varphi$, voor alle b.

– *Een formule heet* **waar** *of* **logisch geldig** *of* **universeel geldig***, notatie*

$$\models \varphi,$$

als deze waar is in alle modellen.

– *Een formule φ heet* **onwaar** *(Eng.:* **false***) met betrekking tot $\mathcal{M}$ als $\mathcal{M}, b \not\models \varphi$, voor alle b.*

– *Een formule φ heet* **onwaar** *of een* **contradictie** *als deze onwaar is in alle modellen.*

– *Een formule φ heet* **contingent** *(Eng.: id.) als deze waar noch onwaar is.*

Contingent betekent zoveel als "het hangt er van af".

Opgave 11.9 Bewijs:

1. Contradictie $\Leftrightarrow$ onvervulbaar.
2. Vervulbaar $\Leftrightarrow$ geen contradictie.
3. Contingent $\Rightarrow$ vervulbaar.
4. Waar $\Rightarrow$ vervulbaar.
5. Contradictie $\Rightarrow$ niet waar.
6. Er geldt niet: niet waar $\Rightarrow$ contradictie.

Opgave 11.10 Bewijs: een formule is contingent als en slechts als er $\mathcal{M}_1, b_1$ en $\mathcal{M}_2, b_2$ zijn, zó dat

$$\mathcal{M}, b_1 \models \varphi \text{ en } \mathcal{M}, b_2 \not\models \varphi$$

Einde opgave.

We schakelen nu voor het gemak even over van willekeurige formules naar *gesloten* formules (predikatenlogische zinnen).

Definitie 11.4 *Twee gesloten formules die waar zijn in precies dezelfde modellen heten* **logisch equivalent**.

De formules $\forall x Px$ en $\neg\exists x\neg Px$ zijn bijvoorbeeld logisch equivalent. Men noemt een model waarin φ waar is wel *een model van* φ. Dus: twee zinnen φ en ψ zijn logisch equivalent wanneer ze dezelfde modellen hebben. We kunnen nu ook spreken over de klasse van modellen van een *verzameling* zinnen. MOD(Γ) duidt bijvoorbeeld de klasse aan van alle modellen voor de taal in kwestie waarin alle zinnen uit de zinnenverzameling Γ waar zijn.

Opgave 11.11 Geef een formele definitie van logische equivalentie. De definitie zal er zo uitzien:

"Twee formules $\varphi, \psi \in \ldots$ heten *logisch equivalent* als

voor alle modellen $\mathcal{M}$: $\ldots \Leftrightarrow \ldots$"

11.3 Substituties

We hebben hierboven bij het uitleggen van het semantische effect van kwantificatie gebruik gemaakt van bedelingen voor variabelen. Deze bedelingen waren nodig omdat—in het algemeen—niet elk ding in het domein van het model een naam hoeft te hebben. Wanneer wel elk ding in het domein van het model een naam heeft, dat wil zeggen wanneer er voor elk object $d \in D$ een constante c uit de taal is zo dat $I(c) = d$, dan kunnen we de werking van kwantificatie uitleggen zonder gebruik te maken van bedelingen.

Terwille van die uitleg maken we eerst een nieuwe notatieafspraak. We spreken af dat $[c/v]\varphi$ betekent: het resultaat van vervanging (substitutie) van alle *vrije voorkomens* van variabele v in φ door constante c.

Spreek "$[c/v]\varphi$" uit als: "φ met c voor v".

Enkele voorbeelden:

$[a/x]Rxy$	staat voor	Ray
$[a/x]Rxx$	staat voor	Raa
$[a/x](Rxy \wedge \exists x Px)$	staat voor	$Ray \wedge \exists x Px$
$Rxy \wedge [a/x]Px$	staat voor	$Rxy \wedge Pa$.

Opgave 11.12 Welke formules worden aangeduid met:

1. $[a/x](\exists x Rxx \wedge Px)$
2. $[b/y](\exists x Rxy \wedge Px)$
3. $[a/x]\exists x \exists y Rxy \rightarrow Px$
4. $[b/y]\exists x \exists y (Rxy \wedge Py)$
5. $[b/y](\exists x \exists y Rxy \wedge Py)$
6. $[a/x]\forall x \forall y Rxy \rightarrow Px$
7. $[a/x](\forall x \forall y Rxy \rightarrow [b/x]Px)$
8. $\exists x Px \wedge [a/x]\exists y Rxy$.

Opgave 11.13 1. Geef een recursieve definitie van de operatie $[e/v]$ op termen. (Aanwijzing: splits uit naar drie gevallen: constante, variabele, samengesteld.)

2. Geef een recursieve definitie van de operatie $[e/v]$ op formules. (Hou er rekening mee dat een substitutie gebonden variabelen ongemoeid laat.)

Voor atomaire formules die vrije variabelen bevatten moeten we nu een afspraak maken die de bedelingen omzeilt. Dit is alleen een kwestie van een knoop doorhakken: we spreken gewoon af dat een atomaire formule waarin variabelen voorkomen waar is in een model $\mathcal{M}$ wanneer er constanten voor die variabelen kunnen worden gesubstitueerd met als resultaat een atomaire formule met alleen constanten die waar is in het model $\mathcal{M}$. Die substitutie moet natuurlijk wel *uniform* gebeuren, dat wil zeggen dezelfde constante moet worden gesubstitueerd voor *alle* voorkomens van een bepaalde variabele, zodat substitutie van a voor x in $Gxyx$ oplevert: $Gaya$. Het kwantor-geval gaat nu als volgt:

$$V_{\mathcal{M}}(\exists v \varphi) = 1 \Leftrightarrow V_{\mathcal{M}}([c/v]\varphi) = 1$$

voor tenminste één constante c van de taal

$$V_{\mathcal{M}}(\forall v\varphi) = 1 \Leftrightarrow V_{\mathcal{M}}([c/v]\varphi) = 1$$
voor elke constante c van de taal

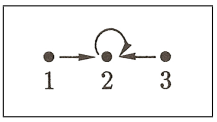
Je ziet het, de bedelingen zijn nu overbodig geworden.

Wanneer we ons beperken tot modellen waarin elk object een naam heeft, terwijl bovendien die modellen *eindig* zijn, dan kan het *nog* simpeler. We mogen nu wel aannemen dat de taal ook maar eindig veel constanten heeft, anders zou er minstens één object zijn dat oneindig veel namen heeft, en dat is in een niet-theologische context een beetje overdreven. Dus: de taal heeft een verzameling constanten $c_1,\ldots,c_n$. De kwantoren kunnen nu volledig *geëlimineerd* worden, en wel als volgt.

– Vervang $\forall v\varphi$ door $[c_1/v]\varphi \wedge [c_2/v]\varphi \wedge \cdots \wedge [c_n/v]\varphi$.

– Vervang $\exists v\varphi$ door $[c_1/v]\varphi \vee [c_2/v]\varphi \vee \cdots \vee [c_n/v]\varphi$.

Dit ziet er misschien een beetje te abstract uit; daarom snel een simpel voorbeeld. Beschouw de volgende structuur:



We interpreteren R weer als de pijlrelatie, en we nemen aan dat de taal drie namen heeft, waarbij a object 1 benoemt, b object 2, en c object 3. De volgende bewering is waar in het model: $\exists xRxx$. We kunnen nu echter evengoed zeggen: $Raa \vee Rbb \vee Rcc$. Dit drukt hetzelfde uit. De volgende bewering, die *niet* waar is in het model, kan eveneens worden geparafraseerd met behulp van constanten: $\forall xRxx$. De parafrase wordt: $Raa \wedge Rbb \wedge Rcc$. Tot slot parafraseren we de formule $\forall x\exists yRxy$ (waar in het model). Eerst elimineren we de universele kwantor: $\exists yRay \wedge \exists yRby \wedge \exists yRcy$. Nu nog de existentiële kwantor:

$$(Raa \vee Rab \vee Rac) \wedge (Rba \vee Rbb \vee Rbc) \wedge (Rca \vee Rcb \vee Rcc).$$

Je ziet: de beperking tot eindige modellen maakt van de predikatenlogica weer een soort van propositielogica. Dit heeft een belangrijke theoretische consequentie. Net als de propositielogica is de aldus 'gekortwiekte' predikatenlogica *beslisbaar*: er kan door het volgen van een mechanische procedure worden uitgemaakt of zekere formule logisch geldig is. Dit is een eigenschap die de predikatenlogica in het algemeen *niet* heeft (zie § 14.3).

11.4 Predikatenlogica met identiteit

Een van de manieren om de uitdrukkingskracht van de predikatenlogica belangrijk te vergroten is door het invoeren van het *identiteitsteken* (Eng.: *identity sign*, *equality sign*). Voor identiteit gebruiken we het teken =. Dit is een tweeplaatsig predikaatsymbool, maar voor het gemak schrijven we $a = b$ in plaats van $= ab$. Het identiteitsteken heeft een vaste interpretatie, die gegeven wordt door de volgende waarheidsclausule (we stappen weer over naar het meest algemene geval, en maken dus gebruik van bedelingen):

– $V_{\mathcal{M},b}(t_1 = t_2) = 1 \iff W_{\mathcal{M},b}(t_1) = W_{\mathcal{M},b}(t_2)$.

Volgens deze definitie is $a = b$ waar precies in die gevallen waarin a en b hetzelfde object benoemen.

Hoe handig het invoeren van het identiteitsteken is blijkt uit de volgende parafrasen van Nederlandse zinnen:

'Annie houdt alleen van Bhagwan' : $\forall x(Hax \equiv x = b)$

'Alleen Bhagwan houdt van Annie' : $\forall x(Hxa \equiv x = b)$

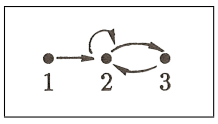
'Piet houdt van iedereen behalve van Bhagwan' : $\forall x(Hpx \equiv \neg x = b)$

'Iedereen houdt van Bhagwan behalve Piet' : $\forall x(Hxb \equiv \neg x = p)$.

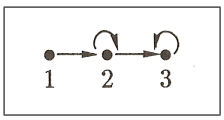
Opgave 11.14 Vertaal nu zelf: 'Piet houdt van Annie, maar Annie houdt van een ander'.

Nog iets moeilijker is: 'Alleen Piet houdt alleen van Annie'. De vertaling wordt: $\forall x(\forall y(Hxy \leftrightarrow y = a) \leftrightarrow x = p)$.

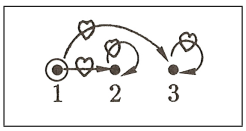
Opgave 11.15 Ga na of deze laatste formule waar is in het volgende model (neem aan dat p wordt geïnterpreteerd als individu 1, en a als individu 2, terwijl individu 3 naamloos is):



Opgave 11.16 Zelfde vraag voor het volgende model (de interpretatie van de constanten blijft hetzelfde):



Opgave 11.17 Beschouw het volgende model:



De pijl $\heartsuit\!\!\rightarrow$ geeft aan wie van wie houdt; $\bigcirc$ geeft aan wie zielig is. De taal die we bekijken heeft een eenplaatsige predikaatletter Z voor 'zielig zijn' en een tweeplaatsige predikaatletter H voor 'houden van'.

Bepaal voor elk van de nu volgende formules eerst of de formule waar is in het model, en geef vervolgens een parafrase van de formules in het Nederlands.

1. $\forall x\exists yHxy$
2. $\forall x\exists y(\neg x = y \wedge Hxy)$
3. $\forall x(\neg Hxx \rightarrow Zx)$

4. $\exists x \exists y(Hxy \wedge \neg x = y \wedge Zx)$
5. $\forall x(Hxx \rightarrow \neg \exists y(\neg x = y \wedge Hxy)$
6. $\forall x \forall y((Hxy \wedge \neg x = y) \rightarrow Zx)$.

Hoe krachtig = is als uitdrukkingsmiddel blijkt uit het feit dat we nu ook kunnen *tellen*: 'Annie houdt van minstens twee jongens' wordt:

$$\exists x \exists y(\neg x = y \wedge Jx \wedge Jy \wedge Hax \wedge Hay).$$

Wat 'Annie houdt van minstens vierentwintig jongens' wordt, kun je nu zelf bedenken. Dat het er knap ingewikkeld gaat uitzien blijkt wel uit het feit dat 'er zijn minstens drie x zo dat Px' al niet korter kan dan (we korten $\neg x = y$ af tot $x \neq y$):

$$\exists x \exists y \exists z(x \neq y \wedge x \neq z \wedge y \neq z \wedge Px \wedge Py \wedge Pz).$$

'Er is hoogstens één x zo dat Px' wordt:

$$\forall x \forall y((Px \wedge Py) \rightarrow x = y).$$

Heel fraai is, dat we nu *uniciteit* kunnen uitdrukken. 'Er is precies één x zo dat Px' wordt:

$$\exists x(Px \wedge \forall y(Py \rightarrow y = x)),$$

of, in een iets beknoptere, maar equivalente formulering:

$$\exists x \forall y(Py \leftrightarrow x = y).$$

Bertrand Russell heeft voorgesteld om deze truc te gebruiken om het *bepaald lidwoord* te parafraseren. Zijn beroemde *descriptietheorie* komt neer op het volgende. Lees "*De* koning toornt" als: "Er is precies één x die koning is, en x toornt". Predikaatlogisch wordt dit:

$$\exists x(Kx \wedge \forall y(Ky \rightarrow y = x) \wedge Tx)$$

of equivalent:

$$\exists x(\forall y(Ky \leftrightarrow y = x) \wedge Tx).$$

Je ziet het: het zinsdeel *de koning* is in de parafrase niet meer als zelfstandig onderdeel terug te vinden. Datzelfde gold trouwens ook al voor nominale constituenten als *elke jongen* en *sommige meisjes*, die als sneeuw voor de zon verdwijnen bij een vertaling naar de predikatenlogica. Dit heet: *contextuele eliminatie* van syntactische constituenten.

Russell paste zijn descriptietheorie toe om onderscheid te kunnen maken tussen verschillende lezingen van zinnen waarin uniek bepalende beschrijvingen (Eng.: *definite descriptions*) voorkomen in samenhang met logische operatoren zoals de negatie-operator. De eliminatie-procedure voor descripties is contextueel. Dit houdt in dat we volgens Russell's descriptietheorie "De koning toornt niet" op verschillende manieren kunnen lezen. Afhankelijk van de vraag wat we als de context beschouwen waaruit *de koning* moet worden geëlimineerd krijgen we een andere predikatenlogische formule als vertaling.

De eerste mogelijkheid is: eliminatie met de hele zin als context. We kunnen nu het bereik van de uniek bepalende beschrijving als volgt aangeven:

– De koning toornt niet.

Dit leidt tot de volgende predikatenlogische formule:

$$\exists x(\forall y(Ky \leftrightarrow x = y) \wedge \neg Tx).$$

We kunnen echter ook besluiten alleen het predikaat *toornen* als eliminatie-context te beschouwen. De negatie-operator valt dan buiten de context waaruit de beschrijving wordt geëlimineerd, ofwel: de beschrijving heeft nu kleiner bereik dan de negatie-operator. Schematisch:

– De koning toornt niet.

De formele weergave wordt nu:

$$\neg \exists x(\forall y(Ky \leftrightarrow x = y) \wedge Tx).$$

Opgave 11.18 Beschouw de modellen in Figuur 11.1.

Neem aan dat ♠ de eenplaatsige predikaatletter K interpreteert, en † de eenplaatsige predikaatletter T. Geef voor elk van de twee lezingen van "De koning toornt niet" die Russell onderscheidt aan in welke van deze vier modellen zij waar is.

De descriptietheorie van Russell is overigens niet onweersproken gebleven. Zie voor filosofische kritiek: [Strawson 1950]. Strawson stelt als alternatief voor Russell's theorie een descriptietheorie voor waarin de uniciteit (in een zekere context) van datgene wat door de beschrijving wordt aangeduid niet wordt *uitgedrukt* in de formele weergave, zoals bij Russell gebeurt, maar wordt *verondersteld*. Dit voorstel wordt vaak door taalkundigen als uitgangspunt genomen voor een zogenaamde presuppositietheorie van uniek bepalende beschrijvingen. Een vergelijking tussen de visies van Russell en Strawson vind je in [Gamut 1982, deel 1].

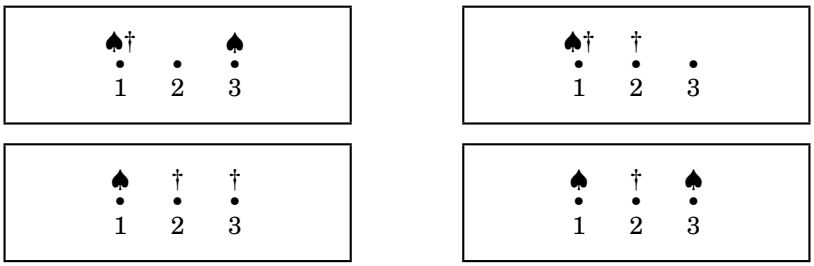
Opgave 11.19 Laat de volgende vertaalsleutel gegeven zijn:

$H(x,y)$: x bezit y
$N(x)$: x is een Nederlander
$F(x)$: x is een fiets
$A(x)$: x is een auto

Vertaal de volgende beweringen in de taal van de predikatenlogica.

1. Iedere Nederlander bezit hoogstens één auto.

2. Iedere Nederlander bezit minstens één fiets en hoogstens twee auto's.



Figuur 11.1: Enige modellen uit de predikaten-logica.

11.5 Functie-symbolen

We hebben gezien hoe de invoering van het identiteitsteken de uitdrukkingskracht van de predikatenlogica vergroot. We gaan nu kort in op een tweede manier om de uitdrukkingskracht van de predikatenlogica te vergroten: het invoeren van *functie-constanten*, namen van een- of meerplaatsige operaties op het individuendomein.

Met functies en meer in het bijzonder met operaties hebben we kennis gemaakt in § 3.2. We recapituleren. Laat een verzameling D gegeven zijn. Een eenplaatsige operatie op D is nu een functie die elementen van D als argumenten neemt (elk element van D kan daarbij als argument optreden), om die elementen af te beelden op elementen van D, de waarden (niet elk element van D hoeft als waarde op te treden). Een tweeplaatsige operatie op D is een functie die *paren* van elementen uit D afbeeldt op elementen van D. Enzovoorts voor drieplaatsige, vierplaatsige,... operaties. Een voorbeeld van een tweeplaatsige operatie op de natuurlijke getallen is de operatie *optellen*. Deze operatie beeldt elk paar van natuurlijke getallen af op een natuurlijk getal, te weten: de som van die getallen.

We breiden nu onze predikatenlogische talen uit met een verzameling van namen voor eenplaatsige operaties, een verzameling van namen voor tweeplaatsige operaties, enzovoorts. Deze uitbreiding betekent dat we nu individuen kunnen aanduiden op nieuwe manieren, en niet meer alleen met behulp constanten of variabelen. Laat f een naam zijn voor een eenplaatsige operatie (we noemen dit: 'een eenplaatsig functiesymbool', of 'een eenplaatsige functieconstante'). Wanneer x en a respectievelijk een variabele en een constante zijn, dan zijn niet alleen x en a manieren om een individu aan te duiden, maar ook fx [of: $f(x)$] en fa [of: $f(a)$] duiden individuen aan. Wanneer h een tweeplaatsig functiesymbool is, dan duidt hax [of: $h(a,x)$] een individu aan.

Hier volgt een schets van de formele uitwerking. De definitie van een *term* wordt nu iets ingewikkelder, en vraagt om een recursieve formulering:

Definitie 11.5 Termen *van een predikatenlogische taal $\mathcal{T}$ met functiesymbolen:*

– *elke variabele of constante is een term;*

– *als g een n-plaatsig functiesymbool is, en $t_1,\ldots,t_n$ zijn termen, dan is $gt_1\ldots t_n$ ook een term;*

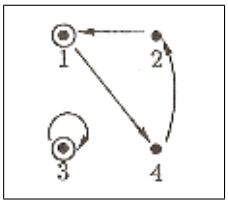
– *niets anders is een term.*

Neem aan dat f een eenplaatsig functiesymbool is, en g een tweeplaatsig functiesymbool; x en y zijn individuele variabelen; a en b zijn constanten. Hier zijn een aantal voorbeelden van termen: x, a, fa, $fffa$, $gxfb$, $fgxfb$, $ggabgffax$, $ffgffafb$, $gagagagagay$. Ga na dat deze voorbeelden aan de definitie voldoen. Soms worden termen waarin functiesymbolen voorkomen met haakjes geschreven: $f(a)$, $g(x,f(b))$, $f(g(x,f(b)))$, enzovoorts. Strikt genomen is dit echter niet nodig, want door de prefix notatie van de functiesymbolen (dat wil zeggen: Poolse notatie) zijn de termen ook zonder haakjes ondubbelzinnig.

De waarde-definitie voor termen moet nu ook worden aangepast. We moeten nu aannemen dat de interpretatiefuncties van modellen voor een predikatenlogische taal $\mathcal{T}$ waarin functiesymbolen voorkomen zo zijn ingericht dat elk van de functiesymbolen wordt afgebeeld op een operatie met het juiste aantal argumenten. Die waarde-definitie maakt gebruik van wat de interpretatiefunctie met de functie-symbolen doet, en dit werkt vervolgens door in de waarheidsdefinitie.

In de volgende opdrachten kun je laten zien dat je intuïtief al weet hoe het zit met waarde van termen en waarheid van formules waarin functiesymbolen voorkomen.

Opgave 11.20 Beschouw het volgende model:



Neem aan dat de pijlen het eenplaatsige functiesymbool f interpreteren (ga na dat dit inderdaad een eenplaatsige operatie is op het domein van de structuur), en de cirkels de eenplaatsige predikaatletter P. De constante a benoemt object 1, b benoemt 2, c benoemt 3, en d benoemt 4. Welk object wordt aangeduid met:

1. fa
2. ffb
3. $fffc$
4. $fffffd$.

Opgave 11.21 De gegevens zijn als in de vorige opdracht. Welke van de volgende formules zijn waar in de structuur:

1. $\exists x Pfx$
2. $\exists x Pfffx$
3. $\forall x(Px \vee Pfffx)$
4. $\exists x(Px \wedge Pfffx)$
5. $\forall x(Pfx \rightarrow \neg Px)$
6. $\forall x(Px \rightarrow (Pfx \rightarrow Pfffx))$.

De algemene formulering van de waarde-definitie $W_{\mathcal{M},b}$ voor een predikatenlogische taal $\mathcal{T}$ met functiesymbolen (waarbij $\mathcal{M}$ een model is voor $\mathcal{T}$ en b een bedeling voor de variabelen uit $\mathcal{T}$ in $\mathcal{M}$) wordt:

- $W_{\mathcal{M},b}(t) = I(t)$ wanneer t een constante is;
- $W_{\mathcal{M},b}(t) = b(t)$ wanneer t een variabele is;
- $W_{\mathcal{M},b}(t) = I(g)(W_{\mathcal{M},b}(t_1),\ldots,W_{\mathcal{M},b}(t_n))$ wanneer t van de vorm $gt_1\ldots t_n$ is.

Deze recursieve waarde-definitie voor termen volgt—zoals je natuurlijk ook verwacht had—weer precies de recursieve definitie van de termen zelf. Aan de waarheidsdefinitie hoeft niets te veranderen.

11.6 Theorieën

Vaak is men geïnteresseerd in de uitspraken die waar zijn in een bepaalde structuur of in een bepaalde klasse structuren. Men kan zelfs proberen een klasse structuren vast te leggen door middel van een verzameling uitspraken die allemaal waar zijn in die structuren. In informaticajargon: men gebruikt de predikatenlogica dan voor de specificatie van een *abstract datatype* (denk bijvoorbeeld aan strings, bomen of Booleans).

In de meer traditionele wiskundige opvatting kunnen verzamelingen zinnen gebruikt worden om een bepaald kennisdomein te axiomatiseren (denk bijvoorbeeld aan de verzamelingenleer, de meetkunde). Vandaar dat in de volgende definitie de term *theorie* wordt gebruikt.

Definitie 11.6 i) *Een theorie T is een verzameling zinnen Γ in een predikaatlogische taal L.*

ii) *Als K de klasse van structuren is die een model zijn voor T, dwz. als $K = \{\mathcal{M} \mid \mathcal{M} \models T\}$, dan zeggen we dat K* geaxiomatiseerd *wordt door T.*

Ad. *i)*: Formeel is T het tupel $T = \langle L, \Gamma\rangle$. Meestal is de taal van T impliciet wel duidelijk, en dan identificeren we T met Γ. Modellen voor Γ noemen we ook modellen voor T.

Wanneer we K opvatten als een abstract datatype, dan kunnen we aan een theorie T die K axiomatiseert dus ook denken als een *specificatie* van het datatype K.

We laten nu een paar belangrijke voorbeelden van theorieën zien. Daarbij nog één opmerking vooraf, over de gebruikte notatie. Volgens de definitie van de syntax van de predikatenlogica bestaat er eigenlijk maar één soort predikaat- en functiesymbolen: met een hoofdletter geschreven en prefix. We hebben hier al tegen gezondigd, en daar zullen we mee doorgaan. Wanneer het de leesbaarheid bevordert maken we zonder schroom gebruik van gewone wiskundige notaties in predikaatlogische formules (zie bv. voorbeeld 11.2 hieronder). De opmerkingen die eerder gemaakt zijn over het principiële onderscheid tussen taal en meta-taal blijven echter onverkort van toepassing.

Voorbeeld 11.1 (Ordeningstheorieën) De taal van de ordeningstheorieën heeft één niet-logisch symbool, de binaire predikaatletter K. De bedoelde interpretatie van $K(x,y)$ is: x is kleiner dan (of gaat vooraf aan) y. Alle ordeningstheorieën bevatten de zinnen (O1) en (O2):

(O1) $\forall x \neg K(x,x)$ (anti-reflexiviteit);

(O2) $\forall xyz((K(x,y) \wedge K(y,z)) \rightarrow K(x,z))$ (transitiviteit).

De axioma's (O1) en (O2) vormen samen de theorie van de *stricte partiële ordeningen*. Door (O3) toe te voegen verkrijgt men de theorie van de *lineaire* (ook wel *totale*) *ordeningen*.

(O3) $\forall xy(x = y \vee K(x,y) \vee K(y,x))$ (samenhang).

Andere belangrijke ordeningsaxioma's zijn:

(O4) $\forall x \exists y K(x,y)$ (voortzettendheid rechts);

(O5) $\forall x \exists y K(y,x)$ (voortzettendheid links);

(O6) $\exists x \forall y(x = y \vee K(x,y))$ (beginpunt);

(O7) $\exists x \forall y(x = y \vee K(y,x))$ (eindpunt);

(O8) $\forall xy(K(x,y) \rightarrow \exists z(K(x,z) \wedge K(z,y)))$ (dichtheid);

(O9) $\forall x(\exists y K(x,y) \rightarrow \exists y(K(x,y) \wedge \neg\exists z(K(x,z) \wedge K(z,y))))$

$\wedge$ $\forall x(\exists y K(y,x) \rightarrow \exists y(K(y,x) \wedge \neg\exists z(K(y,z) \wedge K(z,x))))$ (discreetheid).

Natuurlijk kunnen deze axioma's niet allemaal tegelijkertijd vervuld worden. Verschillende ordeningstheorieën ontstaan door telkens een verschillende greep te doen uit (O4-9).

- T_R, de theorie van de *dichte lineaire ordeningen zonder begin- en eindpunt* heeft de axioma's (O1) t/m (O5) en (O8). Deze theorie beschrijft de ordeningseigenschappen van de rationale en de reële getallen. De bekendste modellen van T_R zijn derhalve $\boldsymbol{Q}^< = \langle \mathbb{Q}, < \rangle$ en $R^< = \langle \mathbb{R}, < \rangle$ met $\mathbb{Q}$ de verzameling rationale getallen en $\mathbb{R}$ de reële getallen.
- T_N, de theorie van de *discrete lineaire ordeningen zonder eindpunt* heeft de axioma's (O1) t/m (O4), (O6) en (O9). Deze theorie beschrijft de ordeningseigenschappen van de natuurlijke getallen. Het bekendste model van TN is $N^< = \langle \mathbb{N}, < \rangle$.

Opgave 11.22 1. Verifieer de ordeningsaxioma's uit de genoemde structuren.

2. Welke van de ordeningsaxioma's zijn waar op $Z^< = \langle \mathbb{Z}, < \rangle$ met $\mathbb{Z}$ de verzameling der gehele (positieve en negatieve) getallen?

Het is een opmerkelijk feit dat er geen zin in de eerste orde predikatenlogica bestaat die discrimineert tussen $Q_<$ en $R_<$. Om het verschil tussen deze twee structuren uit te drukken zijn sterkere logische middelen nodig.

Voorbeeld 11.2 (Peano rekenkunde) Met rekenkunde wordt hier bedoeld het rekenen met natuurlijke getallen. Deze theorie draagt gewoonlijk de naam van de 19de eeuwse Italiaanse wiskundige Giuseppe Peano – hoewel deze zelf eigenlijk andere axioma's voorstelde. De theorie P heeft een taal met als niet-logische symbolen een constante 0, een éénplaatsig functiesymbool s, en twee tweeplaatsige functiesymbolen $+$ en $\cdot$ (met infix notatie). De bedoelde interpretatie van de symbolen 0, $+$ en $\cdot$ is hun gebruikelijke rekenkundige betekenis. De functie s staat voor 'successor', de opvolgerfunctie $s : \mathbb{N} \to \mathbb{N} : n \mapsto n+1$. De axioma's van P zijn:

(P1) $\forall x\, \neg(s(x) = 0)$;

(P2) $\forall xy[s(x) = s(y) \to x = y]$;

(P3) $\forall x(x + 0 = x)$, en $\forall xy[x + s(y) = s(x + y)]$;

(P4) $\forall x(x \cdot 0 = 0)$, en $\forall xy[x \cdot s(y) = (x \cdot y) + x]$;

(P5) voor elke formule φ de universele afsluiting van

$$(\varphi(0) \wedge \forall x[\varphi(x) \to \varphi(s(x))]) \to \forall y \varphi(y).$$

(P5) is het axiomaschema van volledige inductie. Voor iedere formule φ ontstaat een echt axioma, het zijn er dus eigenlijk oneindig veel. Met inductie laten bijvoorbeeld de volgende twee formules zich gemakkelijk bewijzen:

$$\forall x(\neg s(x) = x) \tag{1}$$

$$\forall x(\neg x = 0 \to \exists y(x = s(y)) \tag{2}$$

Het bedoelde model van P is de bekende structuur $N = \langle \mathbb{N}, s, +, \cdot, 0 \rangle$. Dit wordt wel het standaardmodel van de rekenkunde genoemd.

Opgave 11.23 1. Schrijf P1 tot en met P5 met infix-notatie en de gebruikelijke functiesymbolen. Dus schrijf $s(x)$ als $x + 1$, schrijf $\neg(x = y)$ als $x \neq y$, etc.

2. Laat zien dat (1) inderdaad volgt uit de axioma's van de theorie P en dus waar zijn in ieder model van P. (Hint: inductie.)

3. Laat zien dat (2) volgt uit de axioma's van P.

Voorbeeld 11.3 (Boole algebra's) Deze theorie kennen we al in informele gedaante als verzamelingenalgebra's. De taal is als de taal van de rekenkunde, behalve dat de successorfunctie s is vervangen door de eenplaats unaire complementfunctie $'$. De axioma's in hun predikaatlogische vorm:

(B1) ongsaxioma's uit $\forall xy(x + y = y + x)$, $\forall xy(x \cdot y = y \cdot x)$

(B2) distributiviteit: $\forall xyz(x + (y \cdot z) = (x + y) \cdot (x + z))$, $\forall xyz(x \cdot (y + z) = (x \cdot y) + (x \cdot z))$

(B3) neutrale elementen: $\forall x(x + 0 = x), x(x \cdot 0' = x)$

(B4) complement: $\forall x(x + x' = 0')$, $\forall x(x \cdot x' = 0)$

Elke Boole algebra $B = \langle B, s, +, \cdot, 0 \rangle$ is een model van deze theorie. (Ongelukkigerwijs komt "B" hier twee keer voor. Het eerste voorkomen staat op de structuur zelf, het tweede staat op het domein in die structuur, bv. $B = \{0, 1\}$, of $B = \{a, b, c, d\}$, of $B = \{1, \ldots, 16\}$. (Men kan bewijzen dat de kardinaliteit van het domein van een Boole algebra altijd een macht van twee is.)

Voorbeeld 11.4 (Binaire bomen) Er zijn verschillende manieren om een datatype van binaire bomen te axiomatiseren. Onderstaande is ontleend aan [Manna *et al.* 1993]. We gebruiken een taal met twee unaire predikaatsymbolen A (voor atoom) en B (voor boom), en verder een binair functiesymbool $\bullet$ dat we infix zullen noteren. Axioma's:

(T1) $\forall x(A(x) \to B(x))$

(T2) $\forall xy[(B(x) \wedge B(y)) \to B(x \bullet y)]$

(T3) $\forall x \neg(A(x) \wedge B(x))$

(T4) $\forall xyzw[x \bullet y = z \bullet w \to (x = z \wedge y = w)]$

(T5) voor elke formule de universele afsluiting van

$$\forall x(A(x) \to \varphi(x)) \wedge \forall xy((B(x) \wedge B(y) \wedge \varphi(x) \wedge \varphi(y)) \to \varphi(x \bullet y)) \to \forall y(B(y) \to \varphi(y)).$$

Het schema (T5) is het principe van inductie over bomen. Je kunt er bijvoorbeeld mee bewijzen dat geen boom gelijk is aan een directe sub-boom:

$$\forall xyz[(B(x) \wedge (x = y \bullet z)) \to (\neg x = y \wedge \neg x = z)], \tag{3}$$

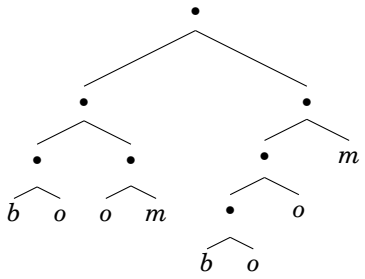
en dat elke boom die geen atoom is een $\bullet$-combinatie is van twee andere bomen:

$$\forall x[(B(x) \wedge \neg A(x)) \to \exists yz((B(y) \wedge B(z) \wedge x = y \bullet z))]. \tag{4}$$

De bedoelde modellen van deze theorie zijn de structuren $B = \langle D, A, B, \bullet \rangle$, met B de verzameling eindige binaire bomen met bladeren uit D. Een voorbeeld van een boom met bladeren uit de verzameling $D = \{b, m, o\}$ is

$$((b \bullet o) \bullet m)) \bullet (((b \bullet o) \bullet o) \bullet m)$$

in boomnotatie:



Opgave 11.24 Bewijs dat de twee uitspraken waarvan hier boven beweerd wordt dat ze gelden in de theorie van binaire bomen inderdaad waar zijn in elk model van deze theorie.

1. Uitspraak (3).
2. Uitspraak (4).

Voorbeeld 11.5 (Stacks) We gebruiken een taal met twee unaire predikaatsymbolen A (voor atoom) en S (voor stack), een binaire functie $Push$, unaire functies Pop en Top en, voor de lege stack, de constante $\emptyset$. Axioma's:

(S1) $S(\emptyset)$

(S2) $\forall xy((A(x) \wedge S(y)) \rightarrow S(Push(x,y))$

(S3) $\forall xy \neg(Push(x,y) = \emptyset)$

(S4) $\forall xyzw(Push(x,y) = Push(z,w) \rightarrow (x = z \wedge y = w))$

(S5) $\forall xy\, Top(Push(x,y)) = x$, $\forall xy\, Pop(Push(x,y)) = y$

(S6) voor elke formule de universele afsluiting van

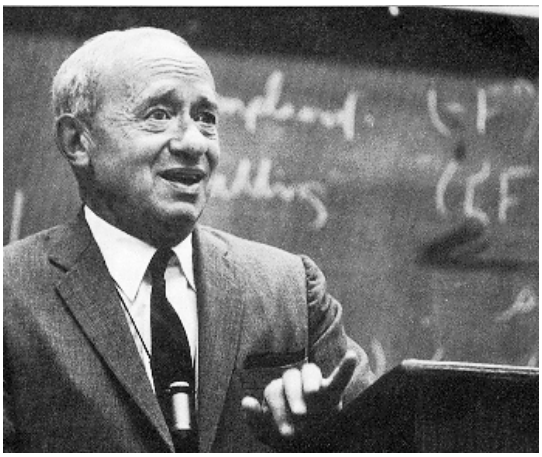
$$(\varphi(\emptyset) \wedge \forall xy((A(x) \wedge S(y) \wedge \varphi(y)) \rightarrow \varphi(Push(x,y)))) \rightarrow \forall y(S(y) \rightarrow \varphi(y)).$$

Opgave 11.25 Bewijs dat uit de axioma's van de theorie van stacks volgt dat

$$\forall x((S(x) \wedge \neg x = \emptyset) \rightarrow Push(Top(x), Pop(x)) = x).$$

Opgave 11.26 Beschouw de structuren $N^< = \langle \mathbb{N}, < \rangle$, $Q^< = \langle \mathbb{Q}, < \rangle$, en $Z^< = \langle \mathbb{Z}, < \rangle$ met $\mathbb{N}$, $\mathbb{Q}$ en $\mathbb{Z}$ respectievelijk de verzamelingen natuurlijke, rationale en gehele getallen en $<$ de "kleiner dan" relatie. De binaire predikaatletter K wordt telkens geïnterpreteerd als $<$.

1. Geef een predikaatlogische zin die waar is in $Z^<$ maar niet in $N^<$.
2. Geef een predikaatlogische zin die waar is in $Z^<$ maar niet in $Q^<$.
3. Geef een predikaatlogische zin die waar is in $N^<$ en $Q^<$ maar niet in $Z^<$.
4. Geef een predikaatlogische zin, geen contradictie, die niet waar is in $N^<$, niet waar is in $Q^<$, en niet waar is $Z^<$.



Alfred Tarski (foto 1971).

11.7 Het substitutielemma

Het substitutielemma is een fundamenteel vehikel (een fundamentele hulpstelling) in praktisch alle logica-bewijzen waarbij syntax aan semantiek moet worden gekoppeld. Het zegt dat substituties van variabelen in formules kunnen worden omgewerkt naar corresponderende bedelingen in de semantiek, en omgekeerd.

Het substitutielemma steunt op zijn beurt weer op het eindigheidslemma voor termen. Het eindigheidslemma voor termen zegt dat waarden van b op variabelen buiten t irrelevant zijn voor de waardebepaling van t. Preciezer zegt het lemma dat de waarde van $W_{\mathcal{M},b}(t)$ wordt bepaald door het gedeelte van b wat op variabelen uit t werkt. (En dat zijn er eindig veel.) Beide lemma's zijn technisch, maar cruciaal in de theorie van de predikatenlogica.

Stelling 11.1 (Eindigheidslemma voor termen) *Als variabele x niet in t voorkomt, dan geldt voor alle bedelingen b en domein-elementen d dat*

$$W_{\mathcal{M},b(x|d)}(t) = W_{\mathcal{M},b}(t).$$

Bewijs: We passen volledige inductie toe naar de termopbouw. De meest eenvoudige termen zijn constanten en variabelen. Deze vormen dus de basis van ons inductieargument.

– Als t een constante is, zeg a, dan werkt aan beide zijden niet de bedeling b op a, maar de interpretatiefunctie I op a. Beide zijden van de gelijkheid reduceren in dit geval dus naar $I(a) \in D$.

– Als t gelijk aan x is, dan zou x voorkomen in t en dat hadden we uitgesloten. Dit geval hoeven we dus niet te behandelen.

– Als t een variabele $y \neq x$ is, dan

$$W_{\mathcal{M},b}(t) = W_{\mathcal{M},b}(y) = b(y).$$

Net zo goed:

$$W_{\mathcal{M},b(x|d)}(t) = W_{\mathcal{M},b(x|d)}(y) = b(x|d)(y) = b(y).$$

Als t gelijk is aan $f(t_1,\ldots,t_n)$, mogen we per inductie aannemen dat het lemma waar is voor alle termen kleiner dan t:

$$\begin{aligned} &W_{\mathcal{M},b(x|d)}(t)\\ &\quad = W_{\mathcal{M},b(x|d)}(f(t_1,\ldots,t_n))\\ &\quad = I(f)(W_{\mathcal{M},b(x|d)}(t_1),\ldots,W_{\mathcal{M},b(x|d)}(t_n)) \end{aligned}$$

Inductiehypothese toepassen:

$$\begin{aligned} &= I(f)(W_{\mathcal{M},b}(t_1),\ldots,W_{\mathcal{M},b}(t_n))\\ &= W_{\mathcal{M},b}(f(t_1,\ldots,t_n))\\ &= W_{\mathcal{M},b}(t). \end{aligned}$$

Omdat hiermee nu alle gevallen behandeld zijn, is het bewijs van het eindigheidslemma voor termen voltooid. □

Het eindigheidslemma voor formules zegt dat alleen de waarden van b op vrije variabelen in φ relevant zijn voor de geldigheid van φ.

Stelling 11.2 (Eindigheidslemma voor formules) *Als variabele x niet vrij voorkomt in φ, dan geldt voor alle bedelingen b en domein-elementen d dat*

$$\mathcal{M}, b(x|d) \models \varphi \iff \mathcal{M}, b \models \varphi.$$

Bewijs: Zoals gebruikelijk bewijzen we dit resultaat met inductie naar de opbouw van de formule. We bewijzen enkele typische gevallen: φ is een atomair predikaat, φ is een conjunctie, φ kwantificeert over een variabele x, en φ kwantificeert over een variabele $y \neq x$.

1. Als φ atomair is, is er een predikaatletter P, en zijn er termen $t_1, \ldots, t_n$ zodanig dat φ gelijk is aan $P(t_1,\ldots,t_n)$.

$$\begin{aligned} &\mathcal{M}, b(x|d) \models P(t_1,\ldots,t_n)\\ &\quad \iff (W_{\mathcal{M},b(x|d)}(t_1),\ldots,W_{\mathcal{M},b(x|d)}(t_n)) \in I(P) \end{aligned}$$

Omdat x niet voorkomt in $P(t_1,\ldots,t_n)$, komt x ook niet voor in elk van de t_i. Op deze termen mogen we dus het eindigheidslemma voor termen toepassen:

$$\begin{aligned} &\iff (W_{\mathcal{M},b}(t_1),\ldots,W_{\mathcal{M},b}(t_n)) \in I(P)\\ &\iff \mathcal{M}, b \models P(t_1,\ldots,t_n). \end{aligned}$$

2. Als φ een conjunctie is dan is φ van de vorm $\varphi_1 \wedge \varphi_2$ en

$$\begin{aligned} &\mathcal{M}, b(x|d) \models \varphi_1 \wedge \varphi_2\\ &\quad \iff \mathcal{M}, b(x|d) \models \varphi_1 \text{ en } \mathcal{M}, b(x|d) \models \varphi_2 \end{aligned}$$

Gebruikmakend van volledige inductie:

$$\begin{aligned} &\iff \mathcal{M}, b \models \varphi_1 \text{ en } \mathcal{M}, b \models \varphi_2\\ &\iff \mathcal{M}, b \models \varphi_1 \wedge \varphi_2. \end{aligned}$$

3. Disjunctie, implicatie en negatie verlopen analoog.

4. Dan zijn we nu aangekomen bij het laatste geval, namelijk

$$\varphi = \forall y \varphi_1.$$

We hebben de volgende feitjes nodig.

i) Er geldt $\mathbf{b(x|d)(x|e)} = \mathbf{b(x|e)}$ voor alle bedelingen b en alle domeinelementen d en e. (Ga na!)

ii) Als $x \neq y$ dan $\mathbf{b(x|d)(y|e)} = \mathbf{b(x|d)(y|e)}$ voor alle bedelingen b en alle domeinelementen d en e. (Ga na!)

Als $y = x$, dan geldt:

$$\begin{aligned} &\mathcal{M}, b(x|d) \models \forall x \varphi_1\\ &\quad \iff \text{voor alle } e \in D : \mathcal{M}, b(x|d)(x|e) \models \varphi_1 \end{aligned}$$

Feitje *i)* gebruiken:

$$\begin{aligned} &\iff \text{voor alle } e \in D : \mathcal{M}, b(x|e) \models \varphi_1\\ &\iff \mathcal{M}, b \models \forall x \varphi_1. \end{aligned}$$

Als $y \neq x$ geldt, dan komt x ook niet vrij voor in φ_1, immers de eerste kwantor is niet verantwoordelijk voor eventuele bindingen van x in φ. Daarom:

$$\begin{aligned} &\mathcal{M}, b(x|d) \models \forall y \varphi_1\\ &\quad \iff \text{voor alle } e \in D : \mathcal{M}, b(x|d)(y|e) \models \varphi_1 \end{aligned}$$

Feitje *ii)* gebruiken:

$$\iff \text{voor alle } e \in D : \mathcal{M}, b(y|e)(x|d) \models \varphi_1$$

Nu kunnen we mooi gebruiken dat x ook niet in φ_1 voorkomt, en vervolgens de inductiehypothese toepassen:

$$\begin{aligned} &\iff \text{voor alle } e \in D : \mathcal{M}, b(y|e) \models \varphi_1\\ &\iff \mathcal{M}, b \models \forall y \varphi_1. \end{aligned}$$

De laatste equivalentie volgt gewoon weer de definitie van de $\forall$-kwantor.

Omdat er verder geen wezenlijk andere gevallen meer te beschouwen zijn, is hiermee het bewijs van het eindigheidslemma voltooid. □

Opgave 11.27 Bewijs het eindigheidslemma voor de volgende typen formules.

i) Voor negaties.

ii) Voor disjuncties.

iii) Voor existentiële formules.

Het substitutielemma is zoals gezegd een fundamenteel vehikel in veel logica-bewijzen. Dit lemma is zo belangrijk omdat het een exclusieve en smalle brug is tussen syntax en semantiek. Meer preciezer functioneert het lemma als spil in alle bekende volledigheidsbewijzen van de predikatenlogica.

We bewijzen eerst het lemma voor termen en dan het lemma voor formules. Als over het substitutielemma zonder meer wordt gesproken, dan wordt het substitutielemma voor formules bedoeld.

Stelling 11.3 (Substitutielemma voor termen) *Als* $W_{\mathcal{M},b}(t) = d$, *dan*

$$W_{\mathcal{M},b}([t/x]t') = W_{\mathcal{M},b(x|d)}(t')$$

Bewijs:

1. Als t' een constante is, zeg a, dan werkt aan beide zijden niet de bedeling b op a, maar de interpretatiefunctie I op a. Beide zijden van de gelijkheid reduceren in dit geval dus naar $I(a) \in D$.

2. Als t' gelijk aan x is, dan

   $$W_{\mathcal{M},b}([t/x]t') = W_{\mathcal{M},b}([t/x]x) = W_{\mathcal{M},b}(t) = d,$$

   de laatste gelijkheid volgt uit de aanname in het lemma. Ook geldt:

   $$W_{\mathcal{M},b(x|d)}(t') = W_{\mathcal{M},b(x|d)}(x) = b(x|d)(x) = d.$$

3. Als t' een variabele $y \neq x$ is, dan

   $$W_{\mathcal{M},b}([t/x]t') = W_{\mathcal{M},b}([t/x]y) = W_{\mathcal{M},b}(y) = b(y).$$

   Net zo goed:

   $$W_{\mathcal{M},b(x|d)}(t') = W_{\mathcal{M},b(x|d)}(y) = b(x|d)(y) = b(y).$$

4. Tenslotte, als t' gelijk is aan $f(t_1,\ldots,t_n)$ is dan

   $$\begin{aligned} &W_{\mathcal{M},b}([t/x]t') \\ &\quad = W_{\mathcal{M},b}([t/x]f(t_1,\ldots,t_n)) \\ &\quad = W_{\mathcal{M},b}(f([t/x]t_1,\ldots,[t/x]t_n)) \\ &\quad = I(f)(W_{\mathcal{M},b}([t/x]t_1),\ldots,W_{\mathcal{M},b}([t/x]t_n)) \end{aligned}$$

   Met inductie:

   $$\begin{aligned} &= I(f)(W_{\mathcal{M},b(x|d)}(t_1),\ldots,W_{\mathcal{M},b(x|d)}(t_n)) \\ &= W_{\mathcal{M},b(x|d)}(f(t_1,\ldots,t_n)) \\ &= W_{\mathcal{M},b(x|d)}(t'). \end{aligned}$$

   Omdat hiermee nu alle gevallen behandeld zijn, is het bewijs van het substitutielemma voor termen voltooid.

□

Nu volgt één van de meest (saaie maar) belangrijke lemma's uit de predikatenlogica:

Stelling 11.4 (Substitutielemma voor formules) *Als* $W_{\mathcal{M},b}(t) = d$, *dan*

$$\mathcal{M},b \models [t/x]\varphi \text{ als en slechts als } \mathcal{M},b(x|d) \models \varphi,$$

mits t vrij voor x in φ.

Merk op dat we nu geen gelijkheid moeten bewijzen maar logische equivalentie.

Bewijs: Zoals gebruikelijk bewijzen we dit resultaat met inductie naar de opbouw van de formule. We bewijzen enkele typische gevallen: φ is een atomair predikaat, φ is een conjunctie, φ kwantificeert over een variabele x, en φ kwantificeert over een variabele $y \neq x$. Het bewijs begint makkelijk, maar op het eind wordt het moeilijker omdat er interactie is tussen variabelen.

1. Als φ atomair is, is er een predikaatletter P, en zijn er termen $t_1, \ldots, t_n$ zodanig dat φ gelijk is aan $P(t_1,\ldots,t_n)$. Er geldt

   $$\begin{aligned} &\mathcal{M},b \models [t/x]P(t_1,\ldots,t_n) \\ &\quad \Longleftrightarrow \mathcal{M},b \models P([t/x]t_1,\ldots,[t/x]t_n) \\ &\quad \Longleftrightarrow (W_{\mathcal{M},b}([t/x]t_1),\ldots,W_{\mathcal{M},b}([t/x]t_n)) \in I(P) \end{aligned}$$

   Gebruikmakend van het substitutielemma voor termen:

   $$\begin{aligned} &\Longleftrightarrow (W_{\mathcal{M},b(x|d)}(t_1),\ldots,W_{\mathcal{M},b(x|d)}(t_n)) \in I(P) \\ &\Longleftrightarrow \mathcal{M},b(x|d) \models P(t_1,\ldots,t_n). \end{aligned}$$

2. Als φ een conjunctie is dan is φ van de vorm $\varphi_1 \wedge \varphi_2$ en

   $$\begin{aligned} &\mathcal{M},b \models [t/x](\varphi_1 \wedge \varphi_2) \\ &\quad \Longleftrightarrow \mathcal{M},b \models [t/x]\varphi_1 \wedge [t/x]\varphi_2 \\ &\quad \Longleftrightarrow \mathcal{M},b \models [t/x]\varphi_1 \text{ en } \mathcal{M},b \models [t/x]\varphi_2 \qquad (!) \\ &\quad \Longleftrightarrow \mathcal{M},b(x|d) \models \varphi_1 \text{ en } \mathcal{M},b(x|d) \models \varphi_2 \\ &\quad \Longleftrightarrow \mathcal{M},b(x|d) \models \varphi_1 \wedge \varphi_2. \end{aligned}$$

   Merk op hoe in bovenstaande herschrijven het patroon van rechtvaardigingen (zie rechterkantlijn) gelijkvormig is: eerst passen we een substitutie toe, dan de definitie van vervulbaarheid, dan de inductie-hypothese (of, in het basisgeval, het substitutielemma voor termen), om dan tenslotte de definitie van vervulbaarheid weer toe te passen, maar dan in omgekeerde vorm.

3. Disjunctie, implicatie en negatie verlopen analoog.

4. Dan zijn we nu aangekomen bij het laatste geval, namelijk

   $$\varphi = \forall y \varphi_1.$$

   Bij dit geval gaat de conditie dat t vrij moet zijn voor x in φ een rol spelen.

   We moeten een aanvullend gevalsonderscheid maken, namelijk het geval waarin x niet vrij voorkomt in φ en het geval waarin x wél vrij voorkomt in φ. Het eerste geval is niet echt interessant, omdat de substitutie dan nergens effectief is. Sterker: voor formules waarin x niet vrij voorkomt, is eenvoudig te bewijzen dat

   $$\begin{aligned} &\mathcal{M},b \models [t/x]\varphi \\ &\quad \Longleftrightarrow \mathcal{M},b \models \varphi \\ &\quad \Longleftrightarrow \mathcal{M},b(x|d) \models \varphi. \end{aligned}$$

   voor willekeurige d. (Eindigheidslemma gecombineerd met feit dat een substitutie geen effect heeft binnen het bereik van een kwantor met dezelfde variabele.) In feite hebben we het grootste deel van dit bewijs hierboven al geleverd. Dit was dus het "oninteressante" geval.

   Interessant wordt het, als x wél vrij voorkomt in φ en er dus iets te substitueren valt. In dit geval geldt:

i) $\mathbf{y} \neq \mathbf{x}$. Immers, als $y = x$ dan zou $\forall y\varphi_1$ gelijk zijn aan $\forall x\varphi_1$ en zou x niet meer vrij voorkomen in φ.

ii) $\mathbf{b(x|d)(y|e)} = \mathbf{b(x|d)(y|e)}$ voor alle bedelingen b en alle domeinelementen d en e. (Ga na! Gebruik daarbij het eerste punt.)

iii) **y komt niet voor in t**. Immers: "t vrij voor x in $\forall y\varphi_1$" betekent dat, als sommige vrije voorkomens van x in $\forall y\varphi_1$ vervangen worden door t, geen enkele vrije variabele van t gebonden wordt. Als y wel zou voorkomen in t, zou deze y in t bij een substitutie t/x gebonden worden door de $\forall y$-kwantor in $\forall y\varphi_1$, wat in tegenspraak is met "vrij zijn voor".

iv) Vanwege het eindigheidslemma geldt voor elke bedeling b en elk domeinelement e (ook $e = d$) dat

$$\mathbf{W}_{\mathcal{M},\mathbf{b(y|e)}}(\mathbf{t}) = \mathbf{W}_{\mathcal{M},\mathbf{b}}(\mathbf{t}).$$

Immers, y komt niet voor in t.

Dit wetende geldt, gebruik makende van *i)*:

$\mathcal{M}, b \models [t/x](\forall y\varphi_1)$
$\iff \mathcal{M}, b \models \forall y([t/x]\varphi_1)$

Definitie van $\forall$-kwantor uitschrijven:

$\iff$ voor alle $e \in D$: $\mathcal{M}, b(y|e) \models [t/x]\varphi_1$

Inductiehypothese:

$\iff$ voor alle $e \in D$: $\mathcal{M}, b(y|e)(x|W_{\mathcal{M},b(y|e)}(t)) \models \varphi_1$

Met *iv)* mogen we $W_{\mathcal{M},b(y|e)}(t)$ vereenvoudigen tot $W_{\mathcal{M},b}(t)$:

$\iff$ voor alle $e \in D$: $\mathcal{M}, b(y|e)(x|W_{\mathcal{M},b}(t)) \models \varphi_1$

Definitie van d gebruiken:

$\iff$ voor alle $e \in D$: $\mathcal{M}, b(y|e)(x|d) \models \varphi_1$

Met *ii)* volgt nu

$\iff$ voor alle $e \in D$: $\mathcal{M}, b(x|d)(y|e) \models \varphi_1$
$\iff \mathcal{M}, b(x|d) \models \forall y\varphi_1$.

De laatste equivalentie volgt gewoon weer de definitie van de $\forall$-kwantor. In het bewijs werd *iii)* impliciet gebruikt, omdat *iv)* gebruik maakt van *iii)*. De observaties *i)-iv)* zijn dus allemaal gebruikt.

Omdat er verder geen wezenlijk andere gevallen meer te beschouwen zijn, is hiermee het bewijs van het substitutielemma voor voltooid. □

Opgave 11.28 Bewijs het substitutielemma voor de volgende typen formules.

i) Voor negaties.

ii) Voor disjuncties.

iii) Voor existentiële formules.

Hoofdstuk 12

Semantische tableaus voor de predikatenlogica

In de propositielogica konden semantische tableaus worden gebruikt om in een eindig aantal stappen de geldigheid van een redenering te testen. Er gold: *"of* een redenering geldig is blijkt in een eindig aantal stappen". In de predikatenlogica liggen de zaken minder eenvoudig. Hier kunnen we ook semantische tableaus gebruiken, maar er is nu geen garantie dat we er in een eindig aantal stappen achterkomen of een redenering geldig is. Wel geldt het volgende: *"als* een redenering geldig is, dan blijkt dat in een eindig aantal stappen". Het verschil is subtiel, maar het zal je hopelijk in de loop van dit hoofdstuk duidelijk worden.

12.1 Standaard tableaus

We introduceren de tableau-methode voor predikatenlogische formules weer aan de hand van voorbeelden.

Voorbeeld 12.1 Geldt $\forall xPx \models Pa$?

Stel van niet: dan moeten we een structuur $\mathscr{M}$ en een bedeling b zien te vinden, zó dat $\mathscr{M}, b \models \forall xPx$ maar $\mathscr{M}, b \not\models Pa$. Nu impliceert de eerste uitdrukking dat $\mathscr{M}, b \models Pa$. Voor het gezochte paar $\mathscr{M}, b$ moet dus zowel $\mathscr{M}, b \models Pa$ als $\mathscr{M}, b \not\models Pa$ gelden. Dat is onmogelijk, dus zo'n paar $\mathscr{M}, b$ bestaat niet. Dus de semantische gevolgtrekking $\forall xPx \models Pa$ is niet te falsifiëren, dus deze is waar. In tableau-vorm:

$$\begin{array}{c} \forall xPx \circ_1 Pa \\ \text{links-}\forall, 1 \\ | \\ Pa \circ \\ \times \end{array}$$

Over de notatie: als naar knopen (elementen in de boom) verwezen word, dan zijn deze knopen genummerd door de scheider $\circ$ in het midden te indexeren met een nummer, bijvoorbeeld i: $\circ_i$. Als we dan later een regel toepassen met verwijzing $i : j$, dan verwijzen we naar formule j in knoop i, waarbij we opnieuw tellen na de scheider. Bijvoorbeeld, als we de regel "rechts-$\exists$" toepassen op $\exists zPz$ in

$$\exists xPx, \forall xyRxy \circ_7 \exists xPx, \forall uvQuv, \exists zPz$$

dan schrijven we: "rechts-$\exists$, 7:3".

Voorbeeld 12.2 Geldt

$$\forall x(Ax \to Bx), \forall x(Bx \to Cx) \models \forall x(Ax \to Cx)?$$

Stel van niet: Om $\forall x(Ax \to Cx)$ onwaar te laten zijn moeten we aannemen dat er minstens één individu, d_1, is waarvoor $Ad_1 \to Cd_1$ onwaar is. De tweede stap is nu gewoon propositioneel-logisch: toepassen van de regel voor $\to$-rechts. Daarna moeten de twee universeel gekwantificeerde formules links nog worden behandeld. Wil een universele bewering waar zijn, dan moet zij zeker gelden voor het individu d_1 dat we in de eerste stap hebben ingevoerd. Dit geeft het volgende tableau:

$$\begin{array}{c} \forall x(Ax \to Bx), \forall x(Bx \to Cx) \circ_1 \forall x(Ax \to Cx) \\ \text{rechts-}\forall, 1{:}3 \\ | \\ \circ_2 Ad_1 \to Cd_1 \\ \text{rechts-}\to, 2 \\ | \\ Ad_1 \circ_3 Cd_1 \\ \text{links-}\forall, 1{:}1 \\ | \\ Ad_1 \to Bd_1 \circ_4 \\ \text{links-}\to, 4 \end{array}$$

$$\begin{array}{cc} Bd_1 \circ & \circ Ad_1 \\ \text{links-}\forall, 1{:}2 & \times \\ | & \\ Bd_1 \to Cd_1 \circ_5 & \\ \text{links-}\to, 5 & \\ \diagup \quad \diagdown & \\ Cd_1 \circ \quad \circ Bd_1 & \\ \times \qquad \times & \end{array}$$

Het tableau sluit: de redenering is geldig. Op deze manier hebben we de geldigheid van Aristoteles' beroemdste syllogisme-figuur, de figuur BARBARA, aangetoond. BARBARA is de naam waarmee de Middeleeuwse scholastici het stramien van de volgende redenering aanduidden:

Alle Grieken zijn mensen.
Alle mensen zijn sterfelijk.

Alle Grieken zijn sterfelijk.

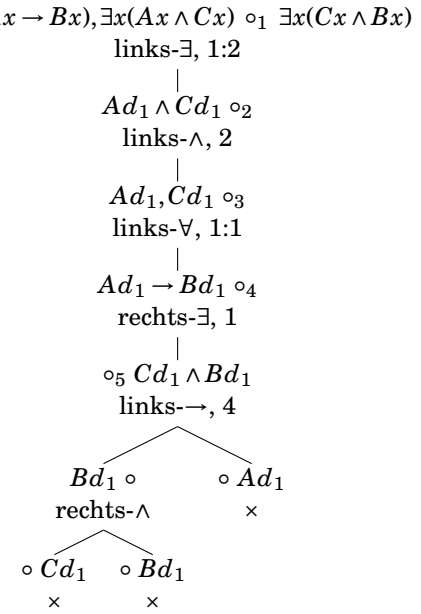
Voorbeeld 12.3 Geldt

$$\forall x(Ax \to Bx), \exists x(Ax \wedge Cx) \models \exists x(Cx \wedge Bx)\ ?$$

We nemen weer aan van niet:

waar	onwaar
$\forall x(Ax \to Bx), \exists x(Ax \wedge Cx)$	$\exists x(Cx \wedge Bx)$

Om de existentiële formule $\exists x(Ax \wedge Cx)$ waar te maken moet voor minstens één object d_1 gelden dat $Ad_1 \wedge Cd_1$. Om de universele formule $\forall x(Ax \rightarrow Bx)$ waar te maken moet voor elk individu in het domein, dus zeker voor d_1, de implicatie gelden. Om de existentiële conjunctie $\exists x(Cx \wedge Bx)$ *onwaar* te maken moet voor *elk individu*, dus zeker ook weer voor d_1, gelden dat de conjunctie onwaar is. Eén en ander leidt tot het volgende tableau:



Ook dit tableau sluit: kennelijk is de redenering geldig. Daarmee is de geldigheid aangetoond van een ander Aristotelisch syllogisme, namelijk het stramien van:

Alle Grieken zijn mensen.
Minstens één Griek is kaal.

Minstens één mens is kaal.

Voorbeeld 12.4 Geldt

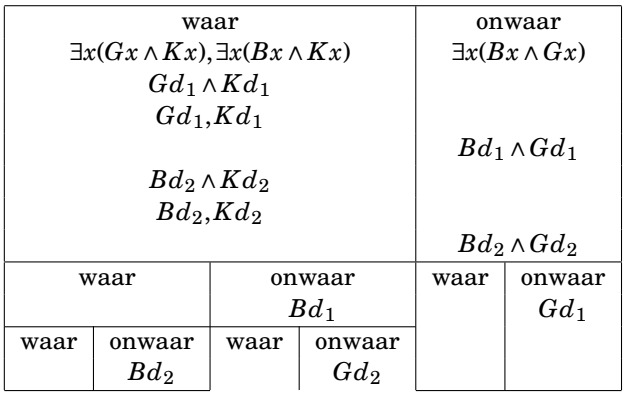
$$\exists x(Gx \wedge Kx), \exists x(Bx \wedge Kx) \models \exists x(Bx \wedge Gx) \text{ ?}$$

Dit is het stramien van de volgende redenering:

Minstens één Griek is kaal.
Minstens één barbaar is kaal.

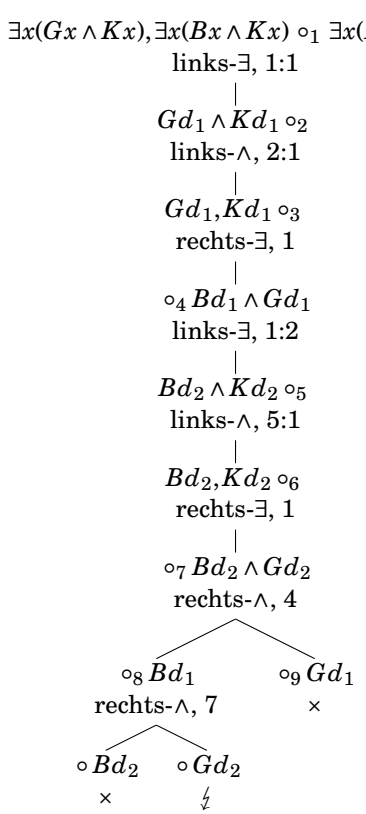
Minstens één barbaar is een Griek.

We nemen weer aan dat de redenering niet geldig is; deze aanname leidt tot het volgende tableau (toelichting volgt):



Twee subtableaus sluiten, maar het derde subtableau blijft open. De redenering is ongeldig.

Hetzelfde tableau in boomvorm:



Een korte toelichting bij de wijze waarop het tableau tot stand is gekomen is op zijn plaats. De volgorde waarin de formules boven in het tableau zijn behandeld, is als volgt. Eerst leidt de behandeling van de eerste existentiële formule links tot invoering van een d_1. Om de existentiële formule rechts onwaar te maken moet wat die formule zegt *niet* opgaan voor d_1, dat wil zeggen: $Bd_1 \wedge Gd_1$ moet onwaar worden. Vervolgens behandelen we de tweede existentiële formule links. Er moet een object d_2 zijn waarvoor de conjunctie $Bd_2 \wedge Kd_2$ waar is. We hadden hier ook het object d_1 voor kunnen nemen, maar dan lopen we het gevaar dat we dat ene object overbelasten door een combinatie van te zware eisen: dat gaat bij voorbeeld zeker fout bij het waar maken van existentiële formules als $\exists xKx \wedge \exists x\neg Kx$. Om dit soort gevaren te omzeilen spreken we af dat we voor elke existentiële formule links een *nieuw* individu zullen introduceren. Nu er een tweede individu is geïntroduceerd moeten we, om $\exists x(Bx \wedge Gx)$ onwaar te doen zijn, ook eisen dat voor d_2 de formule $Bd_2 \wedge Gd_2$ onwaar is. De stappen die daarna volgen zijn propositielogisch.

Uit het open subtableau in bovenstaand tableau kan als volgt een tegenvoorbeeld worden afgelezen. Als we de atomaire formules in dat subtableau langslopen dan zien we dat het subtableau een situatie beschrijft waarin twee dingen allebei de eigenschap K hebben; een ding heeft bovendien de eigenschap B, maar mist G, en het andere heeft G maar mist B. Twee kale individuen, een Griek en een barbaar. Formeel ziet het tegenvoorbeeld tegen de

redenering er als volgt uit: $D = \{1,2\}$, $I(K) = \{1,2\}$, $I(G) = \{1\}$, $I(B) = \{2\}$. Merk op dat strikt genomen de d_1, d_2 in het tableau *individuele constanten* zijn (het zijn immers ingrediënten van formules), terwijl 1 en 2 de objecten in het domein van het tegenvoorbeeld $\mathscr{M} = \langle D, I\rangle$ zijn waarnaar die constanten verwijzen.

Opgave 12.1 1. Teken het Venn-diagram met universum D dat behoort bij het tegenmodel dat is geconstrueerd voor de semantische gevolgtrekking in Voorbeeld 12.4. Geef de verzamelingen $I(K)$, $I(G)$, en $I(B)$ en de individuen $I(d_1)$ en $I(d_2)$ daar in een plek.

2. Leg uit waarom het in dit voorbeeld niet uitmaakt wat de bedeling b is.

3. Is er ook een tegenmodel $\mathscr{M}$ te geven met $|D| = 10$? Zo ja, teken een Venn-diagram. Zo nee, leg uit waarom niet.

4. Is er ook een tegenmodel $\mathscr{M}$ te geven met $d_1 = 10$? Zo, ja beschrijf dat tegenmodel. Zo nee, leg uit waarom niet.

Voorbeeld 12.5 Geldt $\exists xAx \models \forall xAx$?

Nee, want het tableau sluit niet:

waar	onwaar
$\exists xAx$	$\forall xAx$
Ad_1	
	Ad_2

Merk op dat het toepassen van $\forall$-rechts op de d_1 die reeds aanwezig is hier ten onrechte tot sluiting zou hebben geleid. Het volgende is dus FOUT:

waar	onwaar
$\exists xAx$	$\forall xAx$
Ad_1	
	Ad_1

Moraal: $\forall$-rechts en $\exists$-links moeten altijd worden toegepast op een *nieuwe* d_i.

Regels

We sommen de regels voor het behandelen van de kwantoren op:

– $\exists$**-links**: voer een *nieuwe* d_i in, en eis dat $\varphi(d_i)$ waar wordt:

waar	onwaar
$\exists v\varphi$	
$\varphi(d_i)$	

– $\forall$**-rechts**: Voer een *nieuwe* d_i in, en eis dat $\varphi(d_i)$ onwaar wordt:

waar	onwaar
	$\forall v\varphi$
	$\varphi(d_i)$

Merk op dat $\forall$-rechts en $\exists$-links elkaars spiegelbeeld zijn. Net zo zijn $\forall$-links en $\exists$-rechts elkaars spiegelbeeld.

– $\exists$**-rechts**: eis voor alle in het domein aanwezige objecten d_i dat $\varphi(d_i)$ onwaar wordt:

waar	onwaar
	$\exists v\varphi$
	$\varphi(d_i)$

– $\forall$**-links**: eis voor alle in het domein aanwezige objecten d_i dat $\varphi(d_i)$ waar wordt:

waar	onwaar
$\forall v\varphi$	
$\varphi(d_i)$	

De regels $\exists$-rechts en $\forall$-links hebben de vorm van een instructie die terwijl het tableau zich ontwikkelt van kracht blijft; als er later nieuwe elementen worden ingevoerd—ten gevolge van het toepassen van $\forall$-rechts of $\exists$-links—dan moet de instructie ook op die objecten worden toegepast.

Voorbeeld 12.6 Geldt $\exists x\forall yRxy \models \forall y\exists xRxy$?

Ja, kijk maar naar het volgende tableau:

	waar	onwaar
	$\exists x\forall yRxy$	$\forall y\exists xRxy$
$\exists$-links:	$\forall yRd_1y$	
$\forall$-rechts:		$\exists xRxd_2$
$\exists$-rechts:		Rd_1d_2
$\forall$-links:	Rd_1d_2	

$\exists xRxd_2$ moet onwaar worden, en mag dus niet van d_1 gelden: Rd_1d_2 wordt onwaar. $\forall yRd_1y$ moet waar worden en moet dus zeker ook van d_2 gelden: Rd_1d_2 wordt waar. Het tableau sluit.

Net als bij propositielogische tableaus hebben we de volgorde waarin de regels worden toegepast voor een deel zelf in de hand. Bij de propositielogische regels is het slim om toepassen van regels die splitsing veroorzaken zo lang mogelijk uit te stellen. Bij de kwantor-regels is het handig om zo mogelijk $\exists$-links en $\forall$-rechts toe te passen *voor* $\forall$-links en $\exists$-rechts.

Voorbeeld 12.7 Geldt $\forall xAx \models \exists xAx$?

Stel van niet:

waar	onwaar
$\forall xAx$	$\exists xAx$

We zitten nu al meteen in een impasse: $\forall$-links noch $\exists$-rechts kan worden toegepast, want die veronderstellen dat er al individuen zijn ingevoerd. In feite kunnen we hieruit meteen een heel flauw tegenvoorbeeld aflezen, namelijk het model met het lege domein: op het lege domein is $\forall xAx$ waar (omdat er geen individuen zijn geldt elke eigenschap van elk individu), en $\exists xAx$ is onwaar (want deze formule beweert dat er een individu bestaat).

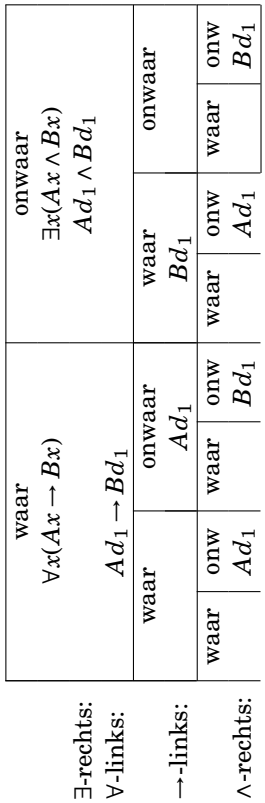
Als we dit soort flauwe tegenvoorbeelden willen uitsluiten moeten we eisen dat de domeinen van onze modellen niet leeg zijn. Maar als we dat afspreken mogen we ook een individu d_1 als 'voorgift' nemen. Het tableau gaat er nu zo uitzien:

	waar	onwaar
	$\forall xAx$	$\exists xAx$
d_1 is voorgift		
$\forall$-links:	Ad_1	
$\exists$-rechts:		Ad_1

Het tableau sluit, en dat wil het volgende zeggen. Als we aannemen dat de domeinen van onze modellen niet leeg zijn, dan is de redenering geldig.

Voorbeeld 12.8 Geldt $\forall x(Ax \to Bx) \models \exists x(Ax \wedge Bx)$?

Als we het lege domein niet uitsluiten zitten we, om dezelfde reden als in het vorige voorbeeld, meteen in een impasse; het model met het lege domein is dan een tegenvoorbeeld tegen de redenering. Sluiten we het lege domein wel uit, dan mogen we weer beginnen met een individu d_1 als voorgift. Het tableau wordt nu:



Drie subtableaus blijven open. Elk van deze subtableaus levert een tegenvoorbeeld. Het subtableau helemaal links:

$$\mathscr{M} = \langle D, I\rangle, \text{ met } D = \{1\}, I(A) = \emptyset, I(B) = \emptyset.$$

Het tweede subtableau van links: zelfde tegenvoorbeeld. Het derde subtableau van links:

$$\mathscr{M} = \langle D, I\rangle, \text{ met } D = \{1\}, I(A) = \emptyset, I(B) = \{1\}.$$

Het bestaan van de tegenvoorbeelden toont aan dat een redenering volgens dit stramien, zoals bij voorbeeld

Alle Grieken zijn sterfelijk.

Minstens één Griek is sterfelijk.

ongeldig is. In de Aristotelische logica wordt deze redenering overigens wel als geldig beschouwd, maar dat komt omdat Aristoteles aanneemt dat geen enkel predikaat een lege extensie heeft.

Opgave 12.2 Bekijk de volgende refutatie van $\forall xRxx \models \forall xyRxy$

(1) $\forall xRxx \circ \forall xyRxy$
links-$\forall$, 1
$Rd_1d_1 \circ$
rechts-$\forall$, 1
(2)$\circ$ $\forall yRd_2y$
rechts-$\forall$, 2
$\circ\, Rd_2d_3$
links-$\forall$, 1
$Rd_2d_2 \circ$
links-$\forall$, 1
$Rd_3d_3 \circ$
↯

1. Specificeer het tegenmodel dat dit tableau voortbrengt.
2. Laat zien dat er een nog korter tableau bestaat voor bovenstaande ongeldige gevolgtrekking.
3. Levert het kortere tableau ook een kleiner tegenmodel op? Zo ja, specificeer dit tegenmodel, zo nee, zoek een korter tableau dat wel een kleiner tegenmodel oplevert.

Voorbeeld 12.9 Geldt $\forall x\exists yRxy \models \forall xRxx$?

Het tableau wordt:

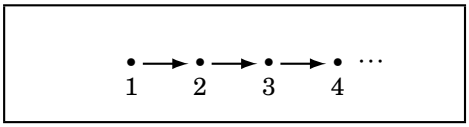
	waar	onwaar
	$\forall x\exists yRxy$	$\forall xRxx$
$\forall$-rechts:		Rd_1d_1
$\forall$-links:	$\exists yRd_1y$	
$\exists$-links:	Rd_1d_2	
$\forall$-links:	$\exists yRd_2y$	
$\exists$-links:	Rd_2d_3	
	$\vdots$	
	enzovoorts	

Kennelijk bevinden we ons in een oneindige lus: $\forall$-links gebiedt dat de formule $\exists yRd_iy$ waar is voor elk individu, en tengevolge van $\exists$-links komen er steeds nieuwe individuen bij. Omdat we weten *hoe* het tableau doorgaat kunnen we zien dat er zeker nooit sluiting zal optreden; er

valt daarom uit het tableau een oneindig tegenvoorbeeld af te lezen:

$$\mathcal{M} = \langle D, I\rangle, \text{ met } D = \{1,2,3,\ldots\},$$
$$I(R) = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle, \ldots\}.$$

In een plaatje:



Het wordt zo langzamerhand tijd om zelf wat te gaan oefenen met het maken van predikatenlogische tableaus.

Opgave 12.3 Test de geldigheid van de volgende redeneringen met tableaus. Geef beschrijvingen van de eventuele tegenvoorbeelden die je vind. Meestal kan door middel van het tekenen van een Venn-diagram en het specificeren van een bedeling waar nodig.

1. $\forall xPxb \models \forall xyPxy$
2. $\forall xyPxy \models \forall xPxb$
3. $\exists xyPxy \models \exists xPxb$
4. $\exists xPxc \models \exists xPxb$

Opgave 12.4 Test de geldigheid van de volgende redeneringen met tableaus. Geef beschrijvingen van de eventuele tegenvoorbeelden.

1. $\forall x(Ax \to Bx), \exists xAx \models \exists xBx$
2. $\models \forall x(((Ax \to Bx) \wedge (Bx \to Cx)) \to (Ax \to Cx))$
3. $\forall x(Ax \to Bx), \neg\exists xBx \models \neg\exists xAx$
4. $\forall x(Ax \to Bx), \neg\exists xAx \models \neg\exists xBx$
5. $\exists x\neg Ax \models \forall x(\forall y(Rxy \to Ax) \to \forall yRxy)$
6. $\exists x\neg Ax \models \forall x(\forall y(Rxy \to Ax) \to \exists y\neg Rxy)$.

Opgave 12.5 Test de geldigheid van de volgende syllogismen met behulp van tableaus:

1. $\forall x(Ax \to Bx), \exists x(Ax \wedge Cx) \models \exists x(Cx \wedge Bx)$
2. $\forall x(Ax \to Bx), \exists x(Ax \wedge \neg Cx) \models \exists x(Cx \wedge \neg Bx)$
3. $\neg\exists x(Ax \wedge Bx), \forall x(Bx \to Cx) \models \neg\exists x(Cx \wedge Ax)$

Opgave 12.6 Bewijs met tableaus:

1. $\neg\exists(Ax \wedge Bx), \exists x(Bx \wedge Cx) \models \neg\forall x(Cx \to Ax)$
2. $\forall x(Ax \to Bx) \vee \forall y(By \to Ay) \models$
$\forall x\forall y((Ax \wedge By) \to (Bx \vee Ay))$
3. $\forall x\forall y((Ax \wedge By) \to (Bx \vee Ay)) \models$
$\forall x(Ax \to Bx) \vee \forall y(By \to Ay)$

Opgave 12.7 Geef met behulp van een tableau een tegenvoorbeeld voor de volgende gevolgtrekking:

$$\forall x\forall y((Ax \wedge By) \to Rxy),$$
$$\forall x\neg Rxx, \forall x(Ax \vee Bx) \not\models \forall x\forall y(Rxy \to Ax)$$

De volgende twee opgaven zijn overgenomen uit een logica-dictaat van Jan Jaspars, UvA/U. Met name de tweede opgave vergt aardig wat van je vaardigheden en inzichten van tableau-technieken.

Opgave 12.8 Bewijs met een tableau dat de 'Quine'-formule onvervulbaar is:

$$\exists x\forall y(Rxy \equiv \neg\exists z(Ry \wedge Rzy))$$

Opgave 12.9 De volgende geldige redeneringen zijn lastig handmatig met een tableau te verifiëren. Als je op de automatische piloot te werk gaat zul je merken dat je snel te lange takken ontwikkelt. Beredeneer daarom eerst zelf dat een tegenmodel onmogelijk is en stuur dan je tableau dezelfde kant op van je eerdere informele redenering.

1. $\forall x\exists yRxy, \forall x\forall y(Rxy \to Ryx),$
$\forall x\forall y\forall z((Rxy \wedge Ryz) \to Rxz) \models \forall xRxx$
2. $\forall x\forall y\forall z((Rxy \wedge Rxz) \to Ryz), \forall xRxx \models$
$\forall x\forall y\forall z((Rxy \wedge Ryz)toRxz)$

Opgave 12.10 Hoe zouden directe tableau-regels voor de existentiële kwantor $\exists!$ ("er is precies één individu zó dat") er uit moeten zien?

Opgave 12.11 De manier van tegenvoorbeelden construeren die we tot nu toe hebben gehanteerd is: in de linkerkolommen van de tableaus aflezen wat verplicht is en ervoor zorgen dat de interpretatie-functie daaraan voldoet. In plaats daarvan zouden we ook een andere strategie kunnen hanteren: in de rechterkolommen van de tableaus aflezen wat verboden is, en de interpretatiefunctie zo inrichten dat de extensies van de predikaten maximaal zijn, i.e., alles omvatten wat *niet* hoeft te worden uitgesloten. Construeer volgens dit principe een tegenvoorbeeld op grond van het tableau uit het bovenstaande voorbeeld.

12.2 Terugkrabbelen

Terugkrabbelen (Eng.: *backtracking*) is een techniek waarbij we in een tableau eerst dingen uitproberen en, als dat mislukt, dan terugkeren naar het punt waarop we besloten hebben "dingen uit te proberen," en het opnieuw proberen met andere variabele-instantiëringen. Het nut van terugkrabbelen is dat we zuiniger omspringen met de al aanwezige constanten door ze proberen te hergebruiken op plekken waar mogelijk onbedoelde afhankelijkheden kunnen worden geïntroduceerd. Door proberen-en-terugkrabbelen kunnen soms kleinere tegenmodellen verkregen worden. We gaan nu in detail uitleggen hoe dat werkt.

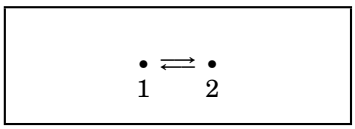
Allereerst de motivering: in feite hadden we bij de laatste redenering ook kunnen volstaan met een *eindig* tegenvoorbeeld. De reden voor het invoeren van telkens een nieuwe d_i bij $\exists$-links en $\forall$-rechts was dat bij het gebruik van een oude d_i ten onrechte sluiting zou kunnen optreden. Maar als we weten dat het tableau toch niet sluit geldt deze overweging niet meer. Om een oneindig tegenvoorbeeld om te zetten in een eindig loont het de moeite om de al aanwezige individuen zoveel mogelijk te hergebruiken door $\exists$-links en $\forall$-rechts eerst op die individuen uit te proberen. Voor ons voorbeeld geeft dit het volgende tableau:

	waar	onwaar
	$\forall x\exists yRxy$	$\forall xRxx$
$\forall$-rechts:		Rd_1d_1
$\forall$-links:	$\exists yRd_1y$	
$\exists$-links:	Rd_1d_2	
$\forall$-links:	$\exists yRd_2y$	
$\exists$-links:	Rd_2d_1	

De eerste keer dat $\exists$-links wordt toegepast moeten we een nieuw individu d_2 invoeren: gebruik van de reeds aanwezige d_1 zou ten onrechte tot sluiting leiden. $\forall$-links is ook voor d_2 van kracht, dus $\exists yRd_2y$ moet waar zijn. Bij de dan volgende toepassing van $\exists$-links zondigen we tegen de boven gegeven regel (verderop volgt een aangepaste versie) en proberen we het reeds aanwezige individu d_1. Het enige motief voor de zonde is: het ontstaan van een oneindig tegenvoorbeeld vermijden. In dit geval hebben we geluk: Rd_2d_1 links plaatsen leidt *niet* tot sluiting. Er zijn nu geen regels meer toepasbaar, en het tableau sluit niet. Het tableau levert ons een eindig tegenvoorbeeld:

$$\mathscr{M} = \langle D, I\rangle, \text{ met } D = \{1,2\}, I(R) = \{\langle 1,2\rangle, \langle 2,1\rangle\}.$$

Een plaatje:

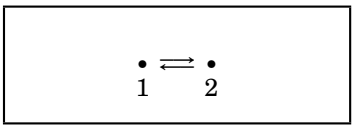


Voorbeeld 12.10 Geldt $\forall x\exists yRxy \models \exists y\forall xRxy$?

Nee, en de tableau-methode levert weer op systematische wijze een tegenvoorbeeld. We gaan uit van de eis dat het domein niet leeg mag zijn, en gebruiken dus d_1 als voorgift. Bij de eerste toepassing van $\exists$-links zou het gebruik van een al aanwezig individu tot sluiting leiden, en daarom mag het niet. We krabbelen terug en proberen opnieuw. Bij de tweede toepassing van $\exists$-links en bij de toepassingen van $\forall$-rechts leidt het gebruik van oude individuen niet tot sluiting, en daarom mag het (vergelijk de opmerking naar aanleiding van het vorige voorbeeld). Zo blijken we in staat te zijn om een eindig tegenvoorbeeld te construeren:

	waar	onwaar
	$\forall x\exists yRxy$	$\exists y\forall xRxy$
d_1 is voorgift		
$\forall$-links:	$\exists yRd_1y$	
$\exists$-rechts:		$\forall xRxd_1$
$\forall$-rechts:		Rd_1d_1
$\exists$-links:	Rd_1d_2	
gebruik van d_1 zou hier ten onrechte tot sluiting leiden		
$\exists$-rechts:		$\forall xRxd_2$
$\forall$-rechts:		Rd_2d_2
$\forall$-links:	$\exists yRd_2y$	
$\exists$-links:	Rd_2d_1	

Het tegenvoorbeeld in een plaatje:



Het proberen te werken met al aanwezige individuen levert de volgende amenderingen in de regels voor $\exists$-links en $\forall$-rechts op:

– $\exists$-links (probeer-versie):

  Plaats $\varphi(d_i)$ links voor een of andere al aanwezige d_i. Als dit lukt zonder dat sluiting optreedt, dan klaar. Als het voor elke al aanwezige d_i tot sluiting leidt, voer dan een nieuwe d_i in en plaats $\varphi(d_i)$ links.

– $\forall$-rechts (probeer-versie):

  Plaats $\varphi(d_i)$ rechts voor een of andere al aanwezige d_i. Als dit lukt zonder dat sluiting optreedt, dan klaar. Als het voor elke al aanwezige d_i tot sluiting leidt, voer dan een nieuwe d_i in en plaats $\varphi(d_i)$ rechts.

Als er bij de probeer-versie van de tableau-regels geen sluiting optreedt bij gebruik van bestaande objecten, en er geldt bovendien dat alle bestaande objecten zijn gebruikt bij toepassingen van $\exists$-rechts en $\forall$-links, dan betekent dit dat we erin geslaagd zijn om een *eindig* tegenvoorbeeld te construeren.

Opgave 12.12 Gebruik de probeer-en-terugkrabbel tableau-methode om tegenmodellen te vinden voor de volgende formules:

1. $\forall x\exists yRxy\exists\forall x\exists yRyx$
2. $\exists x\forall yRxy\forall\exists x\forall yRyx$

Opgave 12.13 Hoe, denk je, zou men de individuele constanten die in predikatenlogische formules kunnen voorkomen moeten behandelen in een semantisch tableau? Geef een regel. Maak vervolgens een semantisch tableau om de predikatenlogische weergave van de volgende redenering te testen, en pas de regel daarin toe:

Geen Griek veracht zichzelf.
Sokrates is een Griek.

Sokrates veracht zichzelf niet.

Opgave 12.14 Geef een regel voor de tableau-behandeling van vrije variabelen.

In de aanhef bij deze paragraaf hadden we gezegd dat de semantische tableau-methode niet in alle gevallen uitsluitsel oplevert over de vraag of een predikatenlogische redenering geldig is. Technisch gezegd: het is geen *beslissingsmethode*. Uit de voorbeelden die we tot nu toe behandeld hebben blijkt dit overigens niet: in alle gevallen waren we in staat om na te gaan of de redenering uit het voorbeeld geldig was of niet. Kennelijk konden zich drie mogelijkheden voordoen:

– sluiting van het tableau,
– geen sluiting, maar er zijn geen regels meer van toepassing,
– het tableau raakt in een oneindige lus.

We zetten deze drie mogelijkheden nog eens op een rijtje:

1. Sluiting geeft aan dat het systematisch zoeken naar een tegenvoorbeeld is mislukt. We weten nu dat de redenering geldig is. Sluiting van een tableau treedt altijd op na een eindig aantal regel-toepassingen.

2. Impasse: geen sluiting, en er kunnen geen regels meer worden toegepast. In dit geval valt er een eindig tegenvoorbeeld af te lezen uit een open tak in het tableau. Overigens is het wel steeds opletten geblazen om te kijken of we in een *echte* impasse zitten. Soms is een impasse slechts schijnbaar omdat we een regeltoepassing vergeten zijn.

3. Een oneindige lus: we kunnen regels blijven toepassen, maar daarbij blijft een en hetzelfde patroon zich herhalen. Als deze situatie zich voordoet kunnen we een oneindig tegenvoorbeeld aflezen.

Vergelijk dit met de propositielogica: daar hadden we alleen de gevallen 1. en 2. Daar konden we het volgende zeggen: de tableaumethode is mechanisch, dus we maken er een computerprogramma van, en we laten de computer draaien tot het tableau af is. Na afloop van het programma kunnen we dan zien of we in situatie 1. of 2. zitten. In het eerste geval was de redenering geldig, in het tweede geval was zij ongeldig. Kortom, in de propositielogica levert de tableau-methode een beslissingsmethode voor geldigheid.

Bij de predikatenlogica zit er hier een addertje onder het gras: omdat ook geval 3. zich kan voordoen heb je kans dat de computer niet stopt. Zolang de computer niet gestopt is weet je niet of de redenering geldig is of niet. Het programma onderbreken om te kijken of het in een lus zit helpt ook niet. Stel dat je het programma na 100 uur draaien onderbreekt, en het blijkt niet in een lus te zitten. We weten dan nog niets over de geldigheid van de redenering; we weten immers niet of het programma niet later, bij voorbeeld na 100 uur en drie minuten draaien, toch nog in een lus zou zijn terechtgekomen, of zou zijn gestopt in geval 1. of 2. Kortom, we weten het volgende, niet meer en niet minder:

> Als een predikatenlogische redenering geldig is, dan blijkt dat in een eindig aantal stappen, want dan zal een computerprogramma dat een tableau maakt voor die redenering na het verstrijken van een eindige hoeveelheid tijd stoppen met een gesloten tableau.

Waarom kunnen we, als we dit weten, dan niet in een eindig aantal stappen nagaan *of* een predikatenlogische redenering geldig is? Het probleem zit hem hier in de omstandigheid dat we, hoewel we weten dat de geldigheid van een geldige redenering in eindig veel stappen aan het licht zal komen, niet van tevoren kunnen zeggen *hoeveel* stappen dat zijn. Zolang de computer niet stopt weten we niets.

12.3 Gelijkheid

Tot slot gaan we ons bekommeren om het speciale relatiesymbool "=". We kunnen hiervoor verschillende tableauregels introduceren. Het volgende stelsel van drie regels, afkomstig van [Jeffrey 1967], is één van de meest intuïtieve benaderingen voor gelijkheid in tableaus.

Voor elke term s mogen we altijd de gelijkheid $s = s$ links opvoeren:

– **=-id**:

waar	onwaar
$s = s$	

Termen waarvan we al weten dat ze gelijk zijn, mogen we vervangen. De vervanging is georiënteerd, dat wil zeggen, vindt plaats in één richting:

– **=-links**:

waar	onwaar
$s = t, \varphi(s)$	
$\varphi(t)$	

– **=-rechts**:

waar	onwaar
$s = t$	$\varphi(s)$
	$\varphi(t)$

Een georiënteerde vervanging worden vaak een *herschrijving* (Eng.: *rewriting*) genoemd.

Redeneren met gelijkheid is in het algemeen bijzonder moeilijk, omdat vaak niet duidelijk is hoe bestaande gelijkheden moeten worden geïnstantieerd (regel =-id), en welke gelijkheden je moet toepassen in welke formules (regels =-links en =-rechts). Met propositietableaus konden we varen op de automatische piloot, bij 1e-ordetableaus al minder (denk aan de keuzevrijheid bij links-$\forall$ en rechts-$\exists$), en bij tableaus met gelijkheid zijn er al helemaal veel keuzemogelijkheden om een tableau voort te zetten. Vaak is dan niet meteen duidelijk welke

volgende stap je moet nemen. Betrek het dus vooral niet op jezelf als je ervaart dat redeneren met gelijkheid moeilijk is. Het is een ingewikkeld onderwerp, niet alleen met tableaus maar ook met andere technieken zoals natuurlijke deductie en resolutie.

Voorbeeld 12.11 Bewijs met tableaus: $s = t \models t = s$.

$$
\begin{array}{c}
(1)\ s = t \circ\ t = s \\
\text{links-=, 1} \\
| \\
t = t \circ \\
\text{rechts-=, 1} \\
| \\
\circ\ t = t \\
\times
\end{array}
$$

Toelichting: steeds wordt de gelijkheid $s = t$ in de eerste knoop links van het cirkeltje toegepast. Je mag hiervoor gerust lezen $s \to t$ en uitspreken: s herschrijft naar t. In de eerste stap word de linker s in de gelijkheid $s = t$ zélf herschreven naar t. In de tweede stap word de rechter s van de gelijkheid $t = s$ herschreven naar t.

De volgende opgave is goed te doen.

Opgave 12.15 Bewijs met tableaus:

1. $\models s = s$.
2. $t = s \models s = t$. Wat is het verschil met het tableau voor in het bovenstaande voorbeeld?
3. $s = t, t = u \models s = u$.

Voorbeeld 12.12 Te bewijzen:

$$\forall x\, f(x) = f(y) \models \forall x\, f(f(x)) = f(x).$$

Dit is redelijk eenvoudig zodra we inzien dat $f(f(x)) = f(x)$ een instantie is van $f(x) = f(y)$:

$$
\begin{array}{c}
\forall x\, f(x) = f(y)\ \circ\ \forall x\, f(f(x)) = f(x) \\
\text{links-}\forall \text{ met } [fx/x, x/y] \\
| \\
f(f(x)) = f(x) \circ \\
\times
\end{array}
$$

Merk op dat in dit bewijs er geen enkele tableau-regel voor gelijkheid is gebruikt.

Voorbeeld 12.13 Bewijs met tableaus:

$$\forall xy\, f(f(x)) = y \models \forall xy\, x = y.$$

Eerst gaan we het rechterlid volledig instantiëren met frisse constanten d en e. Vervolgens proberen we de gelijkheid $d = e$ aan de rechterkant te spiegelen door de gelijkheid aan de linkerkant van knoop 1 een aantal keren "slim" te instantiëren, in dit geval één keer met $[d/x, e/y]$ en één keer met $[d/x, d/y]$. Vervolgens kunnen we de linkerkant van de gelijkheid in knoop 4 herschrijven met behulp van de gelijkheid in knoop 5:

$$
\begin{array}{c}
\forall xy\, f(f(x)) = y\ \circ_1\ \forall xy\, x = y \\
\text{rechts-}\forall\text{, 1, met } [d/x] \\
| \\
\circ_2\ \forall y\, d = y \\
\text{rechts-}\forall\text{, 2, met } [e/y] \\
| \\
\circ_3\ d = e \\
\text{links-}\forall \text{ met } [d/x, e/y] \\
| \\
f(f(d)) = e \circ_4 \\
\text{links-}\forall\text{, 1 met } [d/x, d/y] \\
| \\
f(f(d)) = d \circ_5 \\
\text{links-=, 5 met gelijkheid 4} \\
| \\
d = e \circ \\
\times
\end{array}
$$

Opgave 12.16 Bewijs met tableaus:

1. $\forall x(fffx = x), \forall z(ffz = fz) \models \forall y(fy = y)$
2. $\forall x(fx = f^4x), \forall x(f^5x = f^3x) \models \forall x(f^2x = f^3x)$

Voorbeeld 12.14 Bewijs met tableaus, waarbij "$*$" een binaire functie is:

$$\forall xy(x * y = y) \models \forall xyz(x * (y * z) = (x * y) * z).$$

Eerst passen we drie keer rechts-$\forall$ toe om de kwantoren weg te halen en de formule te instantiëren met concrete constanten. Vervolgens instantiëren we selectief twee keer de universele formule links van $\circ$. Met deze twee instanties kan nu $c * (d * e) = (c * d) * e$ rechts van $\circ$ herschreven worden tot een identiteit $d * e = d * e$. Door dezelfde identiteit met =-id links te introduceren (mag altijd), sluit het tableau:

$$
\begin{array}{c}
\forall xy(x * y = y)\ \circ_1\ \forall xyz(x * (y * z) = (x * y) * z) \\
3\times \text{ rechts-}\forall\text{,1} \\
| \\
\vdots \\
| \\
\circ_2\ c * (d * e) = (c * d) * e \\
\text{links-}\forall\text{,1} \\
| \\
c * (d * e) = d * e \circ_3 \\
\text{links-}\forall\text{, 1} \\
| \\
c * d = d \circ_4 \\
\text{rechts-=, 2 met gelijkheid 4} \\
| \\
\circ_4\ c * (d * e) = d * e \\
\text{rechts-=, 5 met gelijkheid 3} \\
| \\
\circ\ d * e = d * e \\
\text{=-id} \\
| \\
d * e = d * e \circ \\
\times
\end{array}
$$

Opgave 12.17 Bewijs met tableaus:

1. $\forall xy(fx = gy) \models \forall xy(fx = fy)$.

2. $\forall xy(gx = gy) \models \forall xy(g(fx) = g(fy))$.
3. $\forall x(fx = x), \forall xy(f(gx) = f(gy)) \models \forall xy(gx = gy)$.
4. $\forall x(f(gx) = x), \forall xy(g(fx) = g(fy)) \models \forall xy(fx = fy)$.

Opgave 12.18 Bewijs met tableaus, waarbij "$*$" een binaire functie is:

1. $\forall xy(x * y = y) \models \forall xyz((x * y) * z = x * (y * z))$
2. $\forall xyuv(x * y = u * v) \models \forall xyz(x * (y * z) = (x * y) * z)$
3. $\forall xyuv(x * y = u * v) \models \forall xy(x * y = y * x)$
4. $\forall xyuv(x * y = u * v) \models \forall xy(x * x = y * y)$

Automatische stellingenbewijzers die stellingen proberen te bewijzen in equationele logica moeten zeer inventief zijn en op basis van de formule-syntax kunnen beoordelen "welke kant" het uit moet. Daartoe gebruiken dergelijke bewijzers een combinatie van verschillende heuristieken (i.e., systematische probeer-methoden) om te proberen een tableau te sluiten. In het algemeen is equationele logica een moeilijk gebied waar zeker ook op dit moment nog volop onderzoek naar wordt gedaan. Eén van de meest leesbare inleidingen is te vinden in [Burris 1998, Chapter 3: Equational Logic].

12.4 Correctheid

Hier aangekomen heb je hopelijk een goede intuïtie voor de werking van tableaus in de predikatenlogica. Op een gegeven moment rijst dan de vraag of dit systeem correct is. (Bij tableaus in de propositielogica werd deze vraag gesteld op blz. 94); bij natuurlijke deductie in de propositielogica werd deze vraag gesteld op blz. 103.)

We bekijken voor het gemak alleen tableaus zonder gelijkheid. (Gezond- en volledigheid van tableaus *met* gelijkheid is een moeilijker probleem.) Omdat de aanpak dezelfde is als die bij propositionele tableaus, is kan het handig zijn eerst nog even Sectie 8.4 (blz. 8.4) door te nemen.

Gezondheid

Om te laten zien dat een tableaumethode gezond is moet je laten zien dat, als je begint met $\Gamma \circ \varphi$ zó dat $\Gamma \not\models \varphi$, er bij uitbreiding van het tableau altijd tenminste één tak openblijft. (Zie nogmaals Fig. 8.1 op blz. 95.) Voor tableaus in de predikatenlogica is dat niet veel moeilijker dan voor tableaus in de propositielogica.

Opgave 12.19 Bewijs de volgende implicaties.

1. **links-$\forall$, naar beneden** Als

   $$(\forall x\varphi_1), \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$

   open is, dan is elk kind

   $$[t/x]\varphi_1, \varphi_2, \ldots, \varphi_k \circ \psi_1, \ldots, \psi_n$$

   ook open, mits t vrij voor x in φ.

2. **rechts-$\forall$, naar beneden** Als

   $$\varphi_1, \ldots, \varphi_k \circ (\forall x\psi_1), \psi_2 \ldots, \psi_n$$

   open is, dan is het generieke kind

   $$\varphi_1, \ldots, \varphi_k \circ [a/x]\psi_1, \psi_2, \ldots, \psi_n$$

   ook open. Onder 'generiek' verstaan we dat constante a nog niet voorkwam hogerop in de tak.

3. Bewijs de neerwaartse voortplantingsregels voor existentiële kwantificatie.

Volledigheid

Dit is moeilijker. Net zoals bij het volledigheidsbewijs van tableaus in de propositielogica is het in beginsel voldoende te laten zien dat, als je begint met $\Gamma \circ \varphi$, en één tak niet wil sluiten, dat dan $\Gamma \not\models \varphi$. Zoals je kunt nalezen in het bewijs van de volledigheid van de tableaumethode in de propositielogica (blz. 96), schuilt de crux van een volledigheidsbewijs bij tableaumethoden in het kunnen aanwijzen van een model, m, die kan worden geïdentificeerd met de openblijvende tak. In het propositionele geval is dat niet zo moeilijk, immers door decompositie van formules (of, wat op hetzelfde neerkomt, het verdwijnen van connectieven) eindigt een tak altijd, en kunnen we in dat geval het gevraagde model m gewoon aflezen van het blad (i.e., de laatste knoop).

Het probleem bij predikaatlogica is dat takken vanwege links-$\forall$ en rechts-$\exists$ oneindig kunnen zijn, en het dan niet direct duidelijk is of elke oneindige openblijvende tak een model specificeert. Anders gezegd is het niet meteen duidelijk of er in elke tak die niet wil sluiten een model verstopt zit.

Wat we kunnen doen om volledigheid te bewijzen is aannemen dat een openblijvende tak volledig is uitontwikkeld. We nemen dus aan dat in een openblijvende tak elke toepasbare regel ook daadwerkelijk is toegepast. (Ga na waarom zo'n aanname kan worden gemaakt.) Voor universele regels ($\forall$-links en $\exists$-rechts) betekent dat in het bijzonder dat alle instanties van gekwantificeerde formules, en dat zijn er bij universele regels oneindig veel, ook allemaal voorkomen in die tak. Zo'n tak wordt *verzadigd* genoemd, en alle formules die voorkomen in een verzadigde tak wordt een *Hintikka set* genoemd (naar de Finse logicus-filosoof Jaakko Hintikka, 1929-). Een Hintikka-set is de predikaatlogische tegenhanger van een maximaal consistente verzameling. (Een definitie van maximaal consistente verzamelingen is te vinden op blz. 109.)

Intuïtief voel je misschien al aan dat elke Hintikka set een model $\mathscr{M}$ en een bedeling b uitspelt zodanig dat $\mathscr{M}, b \models \Gamma$ maar $\mathscr{M}, b \not\models \varphi$. Als dat niet zo was, dan zou de bijbehorende tak immers sluiten. Dit vermoeden kan worden bewezen. Het bewijs ligt helaas buiten het bereik van deze tekst,—niet omdat het uitzonderlijk moeilijk is, maar omdat ergens een grens moet worden getrokken.

Hoofdstuk 13

Normaalvormen voor de predikatenlogica

Inmiddels weten we dat propositionele formules met behoud van logische equivalentie kunnen worden herschreven naar standaarduitdrukkingen zoals CNF en DNF (Sectie 8.2, blz. 88). We willen dit ook graag kunnen doen voor 1e-orde formules. Het voordeel van normaalvormen is dat ze geschikt zijn voor automatische herkenning en verwerking, nodig voor automatische stellingenbewijzers.

Teneinde de normaalvormen voor de predikatenlogica uit te doeken te doen, moeten we nog wat voorwerk verrichten om er voor te zorgen dat alle variabelen vrij substitueerbaar zijn.

Vrije en gebonden variabelen

We hebben al eerder gezien dat het voorkomen van variabelen in de taal ons af en toe voor extra boekhoudkundig werk stelt. Op bladzijde 118 is al, zij het wat informeel, aangegeven wat met een vrij voorkomen, respectievelijk een gebonden voorkomen van een variabele in een formule wordt bedoeld. Als we dat nog eens netjes willen doen, kunnen we eenvoudig gebruik maken van de inductieve definitie van FOR(L). Sterker nog, omdat variabelen slechts in termen kunnen voorkomen, is het noodzakelijk om eerst precies te maken wat we bedoelen met de variabelen in een term uit TER(L). We definiëren daartoe de functie die alle (vrije) variabelen van een term geeft. We volgen daarbij precies de inductieve definitie van termen:

Definitie 13.1 *De functie*

$$\mathrm{vv} : \mathrm{TER}(L) \to 2^{\mathrm{VAR}}$$

is als volgt gedefinieerd:

- $\mathrm{vv}(x) = \{x\}$ *als x een variabele is;*
- $\mathrm{vv}(c) = \emptyset$ *als c een constante is;*
- $\mathrm{vv}(f(t_1,\ldots,t_n)) = \mathrm{vv}(t_1) \cup \cdots \cup \mathrm{vv}(t_n)$ *als f functiesymbool en $t_1,\ldots,t_n$ termen zijn*

Opgave 13.1 1. Definieer een functie

$$\mathrm{V} : \mathrm{FOR}(L) \to 2^{\mathrm{VAR}}$$

die van een formule de verzameling van alle variabelen die in die formule voorkomen oplevert. Uiteraard verloopt de definitie inductief, naar de opbouw van formules.

2. Definieer een functie $\mathrm{VV} : \mathrm{FOR}(L) \to 2^{\mathrm{VAR}}$ die van een formule de verzameling van alle vrije variabelen die in een formule voorkomen oplevert.

3. Geef de verzameling variabelen van

   (a) $\psi_1 : \forall x R x y \wedge P x$;

   (b) $\psi_2 : R a x \to \forall x \exists y R x y$;

   (c) $\psi_3 : \forall x (R x f(x) \to \exists u \exists x\, u = x)$

4. Geef ook de verzameling vrije variabelen van ψ_1, ψ_2, en ψ_3.

We kunnen nu met behulp van de zojuist geïntroduceerde functies een aantal eigenschappen wat preciezer opschrijven. We kunnen bijvoorbeeld zeggen dat een formule φ een zin is, precies dan als $\mathrm{VV}(\varphi) = \emptyset$. Ook kunnen we nu nauwkeurig een soort 'eindigheidslemma' formuleren en bewijzen. Voor de predikatenlogica zegt zo'n lemma dat, om de waarheid van een formule φ in een model $\mathscr{M}$ en bedeling b te bepalen, we slechts gedeeltelijk hoeven te kijken naar b: namelijk alleen voor zover ze gedefinieerd is op de vrije variabelen van φ.

Definitie 13.2 *Laat $f, g : D \to B$ twee functies zijn. We zeggen dat f en g op $E \subseteq D$ gelijk zijn, notatie $f =_E g$, de beperkingen van f en g op E identiek zijn. In het bijzonder, als $E \subseteq \mathrm{VAR}$ een verzameling variabelen is, en $\mathscr{M} = \langle D, I\rangle$ een structuur, zijn twee bedelingen b en b' gelijk op E, $b =_E b'$ als ze op alle elementen uit E dezelfde objecten uit het domein D toekennen.*

Raadpleeg desnoods even Definitie 3.23 op bladzijde 39.

Opgave 13.2 Laat de structuur $\mathscr{M} = \langle D, I\rangle$ gegeven zijn, $t \in \mathrm{TER}(L)$ en $\varphi \in \mathrm{FOR}(L)$. Bewijs de volgende onderdelen. Bi-implicaties dienen twee kanten op te worden bewezen.

1. Voor alle bedelingen b, b' geldt: als $b =_{\mathrm{vv}(t)} b'$, dan $W_{\mathscr{M},b}(t) = W_{\mathscr{M},b'}(t)$.

2. Voor alle bedelingen b, b' geldt: als $b =_{\mathrm{VV}(\varphi)} b'$, dan $V_{\mathscr{M},b}(\varphi)\varphi = V_{\mathscr{M},b'}(\varphi)$.

3. Voor alle bedelingen b, b' geldt: als $b =_{\mathrm{VV}(\varphi)} b'$, dan $\mathscr{M}, b \models \varphi \Leftrightarrow \mathscr{M}, b' \models \varphi$.

4. Als φ een zin is, dan $\mathscr{M} \models \varphi$ als en alleen als er tenminste één bedeling b is zó dat $\mathscr{M}, b \models \varphi$.

5. Onderdeel 4 geldt in het algemeen niet voor willekeurige formules.

6. Als φ en ψ zinnen zijn, dan geldt $\varphi \models \psi$ als en alleen als voor alle $\mathscr{M}$ geldt dat, als $\mathscr{M} \models \varphi$, dan $\mathscr{M} \models \psi$.

7. Onderdeel 6 geldt in het algemeen niet voor willekeurige formules.

Tot slot maken we eerdere notatie wat preciezer en geven we de notie van deelformule en atomaire formule.

Definitie 13.3 *Als $\{x_1,\ldots,x_n\} \subseteq \mathrm{VV}(\varphi)$, schrijven we ook wel $\varphi(x_1,\ldots,x_n)$.*

Definitie 13.4 *1. Een* atoom *is een formule van de vorm $\bot$, of $t_1 = t_2$ of $P(x_1,\ldots,x_n)$.*

2. De verzameling van deelformules van φ, genoteerd als $\mathrm{SUB}(\varphi)$*, is als volgt gedefinieerd:*

(a) $\{\varphi\}$ als φ atomair is.

(b) $\{\varphi\} \cup \mathrm{SUB}(\psi)$ als $\varphi = \neg\psi$ of $\varphi = Qx\psi$, waarbij $Q \in \{\forall,\exists\}$.

(c) $\{\varphi\} \cup \mathrm{SUB}(\psi_1) \cup \mathrm{SUB}(\psi_1)$ als $\varphi = \psi_1 \bullet \psi_2$, waarbij $\bullet \in \{\wedge,\vee,\rightarrow\}$.

3. We zeggen dat ψ een deelformule *van φ is precies dan als $\psi \in \mathrm{SUB}(\varphi)$. Als bovendien geldt dat $\psi \neq \varphi$ dan heet ψ een* echte deelformule *van φ.*

13.1 Alfabetische varianten

Bekijk de zin

"iedereen heeft wel respect voor iemand". (1)

Met als vertaalsleutel $R(x,y)$ voor "x heeft respect voor y", komen we snel tot een voorstel:

$$\forall x \exists y Rxy \tag{2}$$

Andere voorstellen, zoals bijvoorbeeld

$$\neg\exists x \forall y \neg Rxy \tag{3}$$

zijn weliswaar logisch equivalent, maar er valt over te discussiëren of (3) ook inderdaad een logische vertaling van (1) is. Heel anders ligt dat met alternatieven als

$$\forall u \exists z Ruz \tag{4}$$

of zelfs

$$\forall y \exists x Ryx \tag{5}$$

Van de alternatieven (4) en (5) zien we niet alleen direct dat ze logisch equivalent zijn met (2), ze zijn zelfs bijna identiek. We zullen ze alfabetische varianten van elkaar noemen. We zien dus dat zo'n alfabetische variant ontstaat als we (alle, of enkele) gebonden variabelen in een formule vervangen door nieuwe variabelen. Daardoor verandert de betekenis van zo'n formule niet wezenlijk, zoals direct valt in te zien als we naar de waarheidsdefinitie van de kwantoren kijken in een structuur $\mathscr{M} = \langle D, I\rangle$:

$$\mathscr{M}, b \models \forall x \exists y Rxy \Leftrightarrow$$
voor alle $d \in D$ is er een $e \in D$, zó dat $(d,e) \in I(R)$,

en

$$\mathscr{M}, b \models \forall u \exists z Ruz \Leftrightarrow$$
voor alle $d \in D$ is er een $e \in D$, zó dat $(d,e) \in I(R)$.

We zien dat de variabelen x en y (u en z) hier slechts dienen als een soort van "place-holders": ze verwijzen niet direct naar elementen in het domein D, maar hebben slechts de functie om tijdelijk door kwantoren te worden gebonden, zodat we ze later door elementen uit het domein kunnen vervangen. We gaan nu in deze paragraaf de notie van alfabetische variant preciezer maken; daartoe bewijzen we eerst een algemeen principe van compositionaliteit.

Stelling 13.1 *Zij $\mathscr{M}$ een structuur, en α en β formules waarvoor $\mathscr{M} \models \alpha \equiv \beta$. Stel nu dat ψ is ontstaan uit φ door een aantal voorkomens van α in φ te vervangen door β. Dan geldt $\mathscr{M} \models \varphi \equiv \psi$.*

Bewijs: We merken allereerst op dat uit de definitie van "$\mathscr{M} \models$" volgt dat, voor willekeurige formules φ en ψ geldt dat

$$\mathscr{M} \models \delta \equiv \gamma \Leftrightarrow$$
$$\text{voor alle } b\text{: } (\mathscr{M}, b) \models \delta \Leftrightarrow (\mathscr{M}, b) \models \gamma \tag{6}$$

We behandelen eerst het geval $\alpha = \varphi$. Als we nu om ψ te verkrijgen een aantal voorkomens van α vervangen door β, dan zijn er twee mogelijkheden: of we vervangen die ene α (in dat geval hebben we $\alpha = \varphi$ en $\beta = \psi$ dus $\mathscr{M} \models \varphi \equiv \psi$) of we vervangen niets (dan hebben we $\varphi = \alpha = \psi$), dus ook $\mathscr{M} \models \varphi \equiv \psi$). Vervolgens bekijken we het geval dat α een echte deelformule is van φ, dus $\alpha \neq \varphi$. Nu bewijzen we de stelling met inductie naar de opbouw van φ. We moeten bewijzen dat voor willekeurige b geldt dat $\mathscr{M}, b \models \varphi \equiv \psi$.

i) Inductie-basis. φ is atomair. De enige deelformule α van φ is φ zelf, en dit geval hebben we al beschouwd.

Nu volgt de inductie-stap: de inductiehypothese zegt nu dat, voor $i = 1,2$ als $\mathscr{M} \models \alpha \equiv \beta$ en , en ψ_i is ontstaan uit φ_i door een aantal voorkomens van α in φ_i te vervangen door β, dan geldt $\mathscr{M}, b \models \varphi_i \equiv \psi_i$.

ii) $\varphi = \neg\varphi_1$. We mogen aannemen dat α een deelformule is van φ_1. Stel nu dat ψ is ontstaan door een aantal voorkomens van echte deelformules α in $\neg\varphi_1$te vervangen door β. Dan is $\psi = \psi_1$, waarbij ψ_1 is ontstaan uit φ_1 door een aantal voorkomens van α te vervangen door β. Volgens de inductiehypothese geldt dan dat $\mathscr{M}, b \models \varphi_1 \equiv \psi_1$, dus uit (6) ook

$$\text{voor alle } b\text{: } (\mathscr{M}, b) \models \varphi_1 \Leftrightarrow (\mathscr{M}, b) \models \psi_1 \tag{7}$$

Dan geldt:

$$\begin{aligned}
&\mathcal{M} \models \varphi \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \varphi \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \neg\varphi_1 \\
\Leftrightarrow\ &\text{voor alle } b \text{ geldt niet}: \mathcal{M}, b \models \varphi_1 \qquad \text{vanwege (7)} \\
\Leftrightarrow\ &\text{voor alle } b \text{ geldt niet}: \mathcal{M}, b \models \psi_1 \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \neg\psi_1 \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \psi \\
\Leftrightarrow\ &\mathcal{M} \models \psi
\end{aligned}$$

iii) $\varphi = \varphi_1 \bullet \varphi_2$ met $\bullet \in \{\wedge, \vee, \rightarrow\}$. Het geval $\alpha = \varphi$ is al beschouwd. Als $\alpha \neq \varphi$, dan is α een deelformule van φ_1 of van φ_2. Omdat ψ ontstaan is uit φ door een aantal voorkomens van α door β te vervangen, moet ψ wel van de vorm $\psi = \psi_1 \bullet \psi_2$ zijn, waarbij ofwel ψ_1 is ontstaan uit φ_1 door een aantal vervangingen van α in φ_1 door β, ofwel ψ_2 uit φ_2 op dezelfde manier. Op de formules φ_1 en φ_2 is de inductiehypothese van toepassing, dus we weten dat $\mathcal{M} \models \varphi_1 \equiv \psi_1$ en $\mathcal{M} \models \varphi_2 \equiv \psi_2$, en, met (6), dat

$$\begin{aligned}
&\text{voor alle } b\text{: } (\mathcal{M}, b) \models \varphi_1 \Leftrightarrow (\mathcal{M}, b) \models \psi_1 \\
&\qquad\qquad \text{en } (\mathcal{M}, b) \models \varphi_2 \Leftrightarrow (\mathcal{M}, b) \models \psi_2 \qquad (8)
\end{aligned}$$

Nu geldt:

$$\begin{aligned}
&\mathcal{M} \models \varphi \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \varphi \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \varphi_1 \bullet \varphi_2 \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \varphi_1 \ \mathit{meta}(\bullet)\ \mathcal{M}, b \models \varphi_2 \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \psi_1 \ \mathit{meta}(\bullet)\ \mathcal{M}, b \models \psi_2 \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \psi_1 \bullet \psi_2 \\
\Leftrightarrow\ &\mathcal{M} \models \psi
\end{aligned}$$

Hierbij is $\mathit{meta}(\wedge)$ = "en", $\mathit{meta}(\vee)$ = "of", en $\mathit{meta}(\rightarrow)$ = "impliceert".

iv) $\varphi = \forall x \varphi_1$ We mogen opnieuw aannemen dat α een deelformule is van φ_1. Als we een aantal voorkomens van α vervangen door β, verkrijgen we $\psi = \forall x \psi_1$, waarbij ψ_1 uit φ_1 verkregen is door een aantal voorkomens van α in φ_1 te vervangen door β. Uit de inductiehypothese volgt dan dat $\mathcal{M} \models \varphi_1 \equiv \psi_1$. Nu geldt:

$$\begin{aligned}
&\mathcal{M} \models \varphi \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \varphi \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \forall x \varphi_1 \\
\Leftrightarrow\ &\text{voor alle } b \text{ en voor alle } d \in D\text{: } \mathcal{M}, b(x|d) \models \varphi_1 \\
\Leftrightarrow\ &\text{voor alle } b \text{ en voor alle } d \in D\text{: } \mathcal{M}, b(x|d) \models \psi_1 \\
\Leftrightarrow\ &\text{voor alle } b\text{: } \mathcal{M}, b \models \forall x \psi_1 \\
\Leftrightarrow\ &\mathcal{M} \models \psi
\end{aligned}$$

Het bewijs voor $\varphi = \exists x \varphi_1$ verloopt natuurlijk analoog.

Stelling 13.1 is een formulering van het *compositionaliteitsprincipe*. Geformuleerd op structuurniveau zegt het compositionaliteitsprincipe dat, om de waarheid van een formule op een structuur met een bedeling te bepalen, alleen de waarheid op die structuur van de delen van die formule er toe doen (en dus bijvoorbeeld niet de vorm, of de waarheid van andere formules). Hoe vanzelfsprekend zo'n principe ook lijkt, het vereist nog wel wat zorgvuldigheid om het netjes te bewijzen.

Zoals gezegd hebben we het compositionaliteitsprincipe nu bewezen op structuurniveau. We kunnen ook nog sterkere en een zwakkere varianten formuleren; de volgende stelling leert ons dat Stelling 13.1 in zekere zin de sterkste geldige formulering was.

Stelling 13.2 *Laat α en β predikaatlogische formules zijn; en laat ψ uit φ verkregen zijn door een aantal voorkomens van α in φ te vervangen door β. Dan:*

i) *Voor alle $\mathcal{M}$ geldt: als $\mathcal{M} \models \alpha \equiv \beta$ dan geldt voor alle b: $\mathcal{M}, b \models \varphi \Leftrightarrow \mathcal{M}, b \models \psi$.*

ii) *Het is **niet** zo dat voor alle $\mathcal{M}$ en b geldt: als $\mathcal{M}, b \models \alpha \equiv \beta$ dan $\mathcal{M}, b \models \varphi \equiv \psi$.*

iii) *Voor alle $\mathcal{M}$ geldt: als $\mathcal{M} \models \alpha \equiv \beta$ dan geldt $\mathcal{M}\varphi \Leftrightarrow \mathcal{M}\psi$.*

iv) *Als $\alpha \equiv \beta$, dan geldt $\varphi \equiv \psi$.*

Bewijs:

i) Dit is precies Stelling 13.1.

ii) We geven een tegenvoorbeeld. Neem $\mathcal{M}$ met domein $D = \mathbb{N}$, laat α gelijk zijn aan de formule $x = 0$, laat β gelijk zijn aan $x = 1$, laat $\varphi = \forall x(x = 0 \equiv x = 0)$, en laat $\psi = \forall x(x = 0 \equiv x = 1)$. Neem bedeling b met $b(x) = 2$. Er geldt nu $\mathcal{M}, b \models \alpha \equiv \beta$ (ga na), maar $\mathcal{M}, b \not\models \varphi \equiv \psi$ (ga na).

iii) Bijna direct uit (*i)*). Immers als voor alle b: $\mathcal{M}, b \models \varphi \Leftrightarrow \mathcal{M}, b \models \psi$, dan ook $\mathcal{M} \models \varphi \Leftrightarrow \mathcal{M} \models \psi$. (Het omgekeerde geldt trouwens *niet*.)

iv) Ook uit *i)*. Dit laten we als opgave.

We zullen nu een principe geven waarop we straks compositionaliteit zullen toepassen. Daartoe hebben we eerst een lemma nodig.

Lemma 13.1 *Laat $\mathcal{M} = \langle D, I \rangle$ een structuur zijn met $d \in D$. Zij t een term en y een variabele met $y \notin \mathrm{vv}(t)$. Dan geldt*

$$W_{\mathcal{M}, b(x|d)}(t) = W_{\mathcal{M}, b(y|d)}([y/x]t).$$

Zij φ een formule met $y \notin \mathrm{V}(\varphi)$. Dan geldt

$$M, b(x|d) \models \varphi \quad \Leftrightarrow \quad \mathcal{M}, b(t|d) \models [y/x]\varphi.$$

Bewijs: (i). Met inductie naar t.

– Als t een constante is, zeg c, dan geldt

$$W_{\mathcal{M},b(x|d)}(t) = W_{\mathcal{M},b(x|d)}(c) = I(c),$$
$$W_{\mathcal{M},b(y|d)}([y/x]t) = W_{\mathcal{M},b(y|d)}(c) = I(c).$$

– Als t een variabele is, zijn er twee mogelijkheden: $t = x$ of $t \neq x$. Als $t = x$, dan geldt

$$W_{\mathcal{M},b(x|d)}(t) = W_{\mathcal{M},b(x|d)}(x) = d,$$
$$W_{\mathcal{M},b(y|d)}([y/x]t) = W_{\mathcal{M},b(y|d)}([y/x]x) = W_{\mathcal{M},b(y|d)}(y) = d$$

Als $t \neq x$ dan kan t ook niet gelijk y zijn, omdat $y \notin \mathrm{vv}(t)$. Dus dan is t een van x en y verschillende variabele z, en dan geldt

$$W_{\mathcal{M},b(x|d)}(t) = W_{\mathcal{M},b(x|d)}(z) = b(z)$$
$$W_{\mathcal{M},b(y|d)}([y/x]z) = W_{\mathcal{M},b(y|d)}(z) = b(z).$$

– Stel nu dat het lemma geldt voor de termen $t_1, \ldots, t_n$. Dan geldt

$$W_{\mathcal{M},b(x|d)}(f(t_1,\ldots,t_n)) = f_I(W_{\mathcal{M},b(x|d)}(t_1),\ldots,W_{\mathcal{M},b(x|d)}(t_1))$$

en

$$W_{\mathcal{M},b(y|d)}([y/x]f(t_1,\ldots,t_n)) = f_I(W_{\mathcal{M},b(y|d)}([y/x]t_1),\ldots,W_{\mathcal{M},b(y|d)}([y/x]t_n))$$

Per inductie geldt $W_{\mathcal{M},b(x|d)}(t_i) = W_{\mathcal{M},b(y|d)}([y/x]t_i)$, voor $1 \leq i \leq n$. Omdat substituties en bedelingen geen invloed hebben op functiesymbolen, zijn de laatste waarden ook gelijk.

– Opgave.

Opgave 13.3 Bewijs Lemma 13.1 (13.1).

Stelling 13.3 *Laat φ een formule zijn, en y een variabele met $y \notin \mathrm{V}(\varphi)$. Dan:*

i) $\forall x\varphi \equiv \forall y[y/x]\varphi$

ii) $\exists x\varphi \equiv \exists y[y/x]\varphi$

Bewijs: Laat $\mathcal{M} = \langle D, I\rangle$ een structuur zijn en y een variabele met $y \notin \mathrm{V}(\varphi)$.

$$\begin{aligned} &\mathcal{M}, b \models \forall x\varphi \\ \Leftrightarrow\ &\text{voor alle } d \in D \text{ geldt } \mathcal{M}, b(x|d) \models \varphi \\ \Leftrightarrow\ &\text{voor alle } d \in D \text{ geldt } \mathcal{M}, b(y|d) \models [y/x]\varphi \\ \Leftrightarrow\ &\mathcal{M}, b \models \forall y[y/x]\varphi \end{aligned}$$

waarbij de middelste b-implicatie volgt uit Lemma (13.1).

Opgave 13.4 Laat zien dat we de eis dat $y \notin \mathrm{V}(\varphi)$ **niet** mogen verzwakken tot $y \notin \mathrm{VV}(\varphi)$.

Uit Stelling 13.3 weten we nu dat we in een formule altijd gebonden variabelen mogen herbenoemen. Het resultaat is dan equivalent.

Definitie 13.5 *Laat φ en ψ twee formules zijn.*

1. *We zeggen dat ψ een* herbenoeming *van φ is als er een variabele $y \notin \mathrm{V}(\varphi)$ bestaat zodat φ een deelformule van de vorm $Qx\varphi'$ heeft en ψ is ontstaan uit φ door deze deelformule te vervangen door $Qy[y/x]\varphi'$.*
2. *ψ heet een* alfabetische variant *van φ als er formules $\varphi_1, \varphi_2, \ldots, \varphi_n$ bestaan, $n \geq 2$, zó dat $\varphi_1 = \varphi$, $\varphi_n = \psi$ en voor alle $1 \leq i < n$ is φ_{i+1} een herbenoeming van φ_i.*

Zie Definitie 13.4 op blz. 150 voor de Q-notatie. (Kortweg: voor Q kan iedere kwantor worden gelezen.)

Opgave 13.5 Ga met Definitie 13.5 na of onderstaande paren formules alfabetische varianten van elkaar zijn:

i) $\forall x\exists y(Rxy \to Py)$ en $\forall x\exists u(Rxu \to Pu)$

ii) $\forall x\exists y(Rxy \wedge \neg Py)$ en $\forall y\exists x(Ryx \wedge \neg Px)$

iii) $\forall x\exists x(Rxy \wedge Py)$ en $\forall y\exists x(Rxy \wedge Py)$

iv) $\exists y(Rxy \to Py)$ en $\exists u(Rzu \to Pu)$

v) $\exists x\exists y(Rxy \to Py)$ en $\exists z\exists z(Rzz \to Pz)$

vi) $\forall x\exists y(Rxy \vee Tyx)$ en $\forall u\exists z(Tzu \vee Ruz)$

Stelling 13.4 i) *Als ψ een herbenoeming van φ is, dan geldt $\psi \equiv \psi$.*

ii) *Als ψ een alfabetische variant van φ is, dan geldt $\psi \equiv \psi$.*

Bewijs: (i) Stel dat ψ een alfabetische herbenoeming van φ is doordat er een deelformule $\alpha = \forall x\varphi'$ in φ vervangen is door $\beta = \forall y[y/x]\varphi'$, met $y \notin \mathrm{V}(\varphi)$. Volgens Stelling 13.3 geldt dan $alpha \equiv \beta$. En volgens het principe van compositionaliteit, Stelling 13.2, geldt dan ook $\varphi \equiv \psi$. Het geval $\alpha = \exists x\varphi'$ gaat net zo.

(ii) Laat $\varphi_1, \varphi_2, \ldots, \varphi_n$ bestaan, $n \geq 2$, zodat $\varphi_1 = \varphi$, $\varphi_n = \psi$ en voor alle $1 \leq i < n$ geldt dat φ_{i+1} een herbenoeming van φ_i is. Uit onderdeel (i) volgt dan dat $\varphi_i \equiv \varphi_{i+1}$, en, omdat $\equiv$ een equivalentie-relatie is, geldt $\psi \equiv \psi$.

Gevolg 13.1 *Als φ een formule is, en $X \subset \mathrm{VAR}$ eindig, dan is er een formule ψ waarvoor geldt dat $\varphi \equiv \psi$ en*

$$(\mathrm{V}(\psi) \backslash \mathrm{VV}(\psi)) \cap X = \emptyset$$

Bewijs: *Onmiddellijk uit Stelling 13.4: elke gebonden variabele in φ komt in een deelformule $Qx\varphi'$: die kunnen we vervangen door een deelformule $Qy[y/x]\varphi'$ met $y \notin X$.*

Een toepassing van Gevolg 13.1 is dat het ons helpt het probleem van oneerlijke substituties te omzeilen. Een *oneerlijke substitutie* is een op zich goed uitgevoerde substitutie waarbij vrije variabelen "per ongeluk" gebonden worden.[1] Bekijk bijvoorbeeld

$$\varphi = \forall x(Pxy \to \forall w(Pwx \vee \exists yPzy))$$

[1] De notie van oneerlijke substitutie komt als "niet vrij substituteerbaar" nog aan de orde in het volgende hoofdstuk, zie Def. 14.1 op blz. 162.

De variabele w is niet vrij voor z in φ: in $[w/z]\varphi$ zou een w gebonden voor komen op een plaats waar in φ nog een vrije z voorkwam. Er zou dus een nieuwe binding zijn gecreëerd. Nu zou het kunnen dat we toch graag de substitutie $[w/z]$ willen uitvoeren, omdat φ bijvoorbeeld voorkomt als deelformule in een grotere ψ waarin we die substitutie willen uitvoeren. Wel, dat kan probleemloos, als we eerst φ vervangen door een alfabetische variant φ', bijvoorbeeld

$$\forall u(Pux \vee \exists y Pzy)$$

We weten dat deze formule φ' equivalent is aan φ, en bovendien is w nu wel vrij voor z in φ'.

Een andere toepassing van alfabetische varianten is dat we formules altijd zó op kunnen schrijven, dat elke variabele daarin maar één rol speelt. In volgende definitie is Q weer een kwantor.

Definitie 13.6 *1. Een formule φ heet* eenduidig in de variabele x *als precies één van de volgende eigenschappen geldt:*

(a) $x \notin \mathrm{V}(\varphi)$.

(b) $x \in \mathrm{VV}(\varphi)$ *en φ heeft geen deelformule van de vorm $Qx\psi$.*

(c) $x \notin \mathrm{VV}(\varphi)$ *en φ heeft slechts één deelformule van de vorm $Qx\psi$.*

2. Een formule heet eenduidig *als deze eenduidig is in alle variabelen.*

Een formule heet dus eenduidig in de variabele x als x niet voorkomt, of slechts één rol speelt in die formule. Bijvoorbeeld de rol van een vrije variable. Of de rol van een gekwantificeerde variabele. In het laatste geval mag er dan slechts één x-kwantor zijn.

Opgave 13.6 1. Welke van onderstaande formules zijn eenduidig?

(a) $\forall x(Rxy \to Ryx)$

(b) $\forall x Px \wedge \forall x Qx$

(c) $Px \wedge \forall xy(Ty \vee \exists u Pu)$

(d) $\forall x(Rxy \vee Ryx) \wedge \exists u \exists v(Rux \wedge Rxv)$

(e) $\forall x(\exists y Rxy \to \exists x \exists z Rxz)$

(f) $Px \wedge \exists u Rxu \wedge \neg Qx$

2. Bewijs of weerleg:

(a) Als $\mathrm{V}(\varphi) = \emptyset$ dan is φ eenduidig.

(b) Als $\mathrm{VV}(\varphi) = \emptyset$ dan is φ eenduidig.

Stelling 13.5 *Elke formule is logisch equivalent met een eenduidige formule.*

Bewijs: Zij φ willekeurig. Als φ nog niet eenduidig is, dan is er een variabele $x \in \mathrm{V}(\varphi)$ waarvoor geldt: óf $x \in \mathrm{VV}(\varphi)$ en $x \in \mathrm{V}(\varphi) \backslash \mathrm{VV}(\varphi)$, d.w.z. x komt zowel vrij als gebonden voor; óf $x \notin \mathrm{VV}(\varphi)$ en φ heeft meer dan één deelformule $Qx\varphi'$, met $Q \in \{\forall, \exists\}$. In beide gevallen kunnen we een deelformule $Qx\varphi'$ uit φ verwijderen: laat $x \notin \mathrm{V}(\varphi)$, en laat$\varphi_1$ de formule zijn die uit φ ontstaat door het voorkomen van $Qx\varphi'$ te vervangen door $Qy[y/x]\varphi'$, dan is φ_1 een alfabetische variant van φ en geldt dus $\varphi \equiv \varphi_1$. Bovendien geldt dat φ_1 eenduidig is in y.

Als φ_1 nog steeds niet eenduidig is, dan herhalen we dit proces: uiteindelijk vinden we een φ_n die aan de twee verlangde eigenschappen voldoet. Bovendien is duidelijk dat dit proces stopt: er zijn slechts eindig veel voorkomens van variabelen in een formule, en elke stap uit ons proces vervangt minstens één voorkomen van een variabele door voorkomen van een eenduidige.

Voorbeeld 13.1 Beschouw

$$\varphi = \forall x(Rxy \to \exists z \neg \exists x Rxz) \wedge \forall z(\forall n T(n,x,z) \to \forall z \exists u \exists u P(x,u,u)).$$

De variabele x is vrij in φ en dat moet zo blijven in onze herschrijving. (Immers een andere vrije variable verandert de betekenis van een formule.) De formule φ is nog niet eenduidig in x, dus we zoeken een deelformule $Qx\varphi'$, laten we zeggen $\forall x(Rxy \to \exists z \neg \exists x Rxz)$. Deze kunnen we vervangen door $\forall v(Rvy \to \exists z \neg \exists x Rxz)$. Het eerste resultaat is dus

$$\varphi_1 = \forall v(Rvy \to \exists z \neg \exists x Rxz) \wedge \forall z(\forall n T(n,x,z) \to \forall z \exists u \exists u P(x,u,u)).$$

De formule φ_1 is nog steeds niet eenduidig in x, we kunnen de deelformule $\exists x Rxz$ vervangen door $\exists q Rxq$:

$$\varphi_2 = \forall v(Rvy \to \exists z \neg \exists q Rqz) \wedge \forall z(\forall n T(n,x,z) \to \forall z \exists u \exists u P(x,u,u)).$$

De formule φ_2 is weliswaar eenduidig in x, maar nog niet in z. We kunnen nu $\exists z \neg \exists q Rqz$ vervangen door $\exists s \neg \exists q Rqs$:

$$\varphi_3 = \forall v(Rvy \to \exists s \neg \exists q Rqs) \wedge \forall z(\forall n T(n,x,z) \to \forall z \exists u \exists u P(x,u,u)).$$

De formule φ_3 is nog niet eenduidig in z. We kunnen $\forall z \exists u \exists u P(x,u,u)$ vervangen door $\forall w \exists u \exists u P(x,u,u)$:

$$\varphi_4 = \forall v(Rvy \to \exists s \neg \exists q Rqs) \wedge \forall z(\forall n T(n,x,z) \to \forall w \exists u \exists u P(x,u,u)).$$

In φ_4 tenslotte vervangen we $\exists u \exists u P(x,u,u)$ door $\exists r \exists u P(x,u,u)$ en krijgen

$$\varphi_5 = \forall v(Rvy \to \exists s \neg \exists q Rqs) \wedge \forall z(\forall n T(n,x,z) \to \forall w \exists r \exists u P(x,u,u)).$$

We hebben nu $\varphi \equiv \varphi_5$ en φ_5 is eenduidig.

Opgave 13.7 Geef, voor elk van de formules van Opgave 13.6 een eenduidige formule die ermee equivalent is.

13.2 Prenex-normaalvorm

We zullen nu laten zien dat er voor elke formule φ een formule φ' in normaalvorm bestaat, zodanig dat $\varphi \equiv \varphi'$. We zullen zelfs een algoritme geven dat φ herschrijft tot φ. Preciezer gezegd, zullen we een algoritme geven, zó dat er voor elke formule φ een rijtje $\varphi_1, \ldots, \varphi_n$ gegenereerd wordt, zodanig dat

- $\varphi_1 = \varphi$
- $\varphi_n = \varphi'$
- $\varphi_i \equiv \varphi_{i+1}$ voor $1 \leq i < n$
- er is een basisstap in het algoritme welke φ_i naar φ_{i+1} herschrijft

We zullen eerst die basisstappen beschrijven in Stelling 13.6. Natuurlijk moet zo'n basisstap formules laten overgaan in equivalente formules.

Definitie 13.7 $\forall$ *en* $\exists$ *heten elkaars* duale kwantor. *Dit geven we ook wel aan met een postfix notatie "′": we schrijven dan* $\forall' = \exists$ *en* $\exists' = \forall$.

.

Vanaf nu is Q één van de kwantoren $\forall$ en $\exists$. Merk op dat $Q'' = Q$. We hebben nu een compacte manier om de benodigde equivalenties op te schrijven.

Stelling 13.6 *Laat* φ *en* ψ *formules zijn, met* $x \notin \mathrm{VV}(\psi)$. *Dan geldt:*

i) $\neg Qx\varphi \equiv Q'x\neg\varphi$

ii) $Qx\varphi \bullet \psi \equiv Qx(\varphi \bullet \psi)$, $\bullet = \wedge, \vee$

iii) $\varphi \bullet Qx\psi \equiv Qx(\varphi \bullet \psi)$, $\bullet = \wedge, \vee$

iv) $Qx\varphi \to \psi \equiv Q'x(\varphi \to \psi)$ *(!)*

v) $\varphi \to Qx\psi \equiv Qx(\varphi \to \psi)$

$\mathscr{M} = \langle D, I\rangle$ een model is, dan heten D en I respectievelijk het *domein* en de *interpretatiefunctie* van het model.

We bewijzen *i)* met $Q = \forall$. Zij model $\mathscr{M}$ en bedeling b willekeurig. Dan

$$\begin{aligned}
&\mathscr{M}, b \models \neg\forall x\varphi \\
\Leftrightarrow\ &\text{er geldt niet } \mathscr{M}, b \models \forall x\varphi \\
\Leftrightarrow\ &\text{niet voor alle } d \in D \text{ geldt } \mathscr{M}, b(x|d) \models \varphi \\
\Leftrightarrow\ &\text{er is een } d \in D \text{ zo dat niet } \mathscr{M}, b(x|d) \models \varphi \\
\Leftrightarrow\ &\text{er is een } d \in D \text{ zo dat } \mathscr{M}, b(x|d) \models \neg\varphi \\
\Leftrightarrow\ &\mathscr{M}, b \models \exists x\neg\varphi
\end{aligned}$$

Omdat $\mathscr{M}$ en b willekeurig waren, geldt $\neg\forall x\varphi \equiv \exists x\neg\varphi$. Het bewijs van *i)* met $Q = \exists$ verloopt analoog.

Zie hoe logische tekens in de object-taal in het rechterlid (rechts van de dubbele turnstile) "opschuiven" naar meta-taal in het linkerlid. Omdat de meta-taal gewoon Nederlands is, kunnen we dit Nederlands anders opschrijven en vervolgens weer terug "opschuiven" naar object-taal in het rechterlid. Ga overigens na dat alle implicaties inderdaad bi-directioneel zijn, i.e., we kunnen in het bewijs wel steeds vrolijk bi-implicaties neerpennen, maar dan moet het nog wel waar zijn.

We bewijzen *ii)* met $Q = \forall$ en $\bullet = \wedge$. Zij model $\mathscr{M}$ en bedeling b willekeurig. Dan

$$\begin{aligned}
&\mathscr{M}, b \models \forall x\varphi \wedge \psi \\
\Leftrightarrow\ &\mathscr{M}, b \models \forall x\varphi \text{ en } \mathscr{M}, b \models \psi \\
\Leftrightarrow\ &\mathscr{M}, b \models \forall x\varphi \text{ en } \mathscr{M}, b \models \forall x\psi \qquad \text{immers, } x \notin \mathrm{VV}(\psi) \\
\Leftrightarrow\ &\mathscr{M}, b \models \forall x(\varphi \wedge \psi)
\end{aligned}$$

Omdat $\mathscr{M}$ en b willekeurig waren, geldt $\forall x\varphi \wedge \psi \equiv \forall x(\varphi \wedge \psi)$. De andere drie bewijzen van *ii)* met $Q \in \{\forall, \exists\}$ en $\bullet \in \{\wedge, \vee\}$ verlopen analoog.

iii) net zoals *ii)*. Beter: gebruik *ii)* en het gegeven dat $\bullet \in \{\wedge, \vee\}$ commutatief is, en gebruik compositionaliteit:

$$\varphi \bullet Qx\psi \equiv Qx\psi \bullet \varphi \equiv Qx(\psi \bullet \varphi) \equiv Qx(\varphi \bullet \psi)$$

iv) kan rechtstreeks, maar ook met behulp van al bekende eigenschappen: we weten hoe we "$\to$" kunnen herschrijven in termen van "$\neg$" en "$\wedge$". Dat we dat ook achter een kwantor, of meer algemeen in een deelformule mogen doen, volgt uit het principe van compositionaliteit. We beschouwen het geval $Q = \forall$ en gebruiken *i)* en *ii)*:

$$\begin{aligned}
\forall x\varphi \to \psi &\equiv \neg\forall x\varphi \vee \psi \\
&\equiv \exists x\neg\varphi \vee \psi \equiv \exists x(\neg\varphi \vee \psi) \equiv \exists x(\varphi \to \psi)
\end{aligned}$$

v) als *iv)*.

Opgave 13.8 Geef de bewijzen van Stelling 13.6 die niet volledig gegeven zijn.

Definitie 13.8 *Een formule* φ *staat in* prenex-normaalvorm*, of kortweg* prenex-vorm *als*

$$\varphi = Q_1x_1 \ldots Q_nx_n\psi$$

waarbij $Q_i \in \{\forall, \exists\}$*, voor* $i \leq n$*,* $n \geq 0$*, en in* ψ *komt geen kwantor meer voor.* $Q_1x_1 \ldots Q_nx_n$ *heet* prefix*, en* φ *heet* matrix.

Stelling 13.7 *Elke formule is equivalent met een formule in prenex-normaalvorm.*

Bewijs: Met inductie naar φ.

i) Als φ atomair is, is φ zijn eigen prenex-vorm.

Stel de bewering is waar voor φ_1 en φ_2: zeg hun prenex-vormen zijn respectievelijk

$$\alpha_1 = Q_1^1x_1 \ldots Q_n^1x_n\psi_1$$
$$\alpha_2 = Q_1^2y_1 \ldots Q_m^2y_m\psi_2$$

We mogen er daarbij wel vanuit gaan dat de x_i's niet voorkomen in α_2 en de y_j's niet in α_1: anders kunnen we daarvoor zorgen dankzij Gevolg 13.1.

ii) De prenex-vorm van $\varphi = \neg\varphi_1$ is natuurlijk

$$\varphi \equiv (Q_1^1)'x_1 \ldots (Q_n^1)'x_n \neg\psi_1.$$

iii) Stel nu $\varphi = \varphi_1 \bullet \varphi_2$, met $\bullet \in \{\wedge, \vee\}$. We mochten aannemen dat φ_1 en φ_2 geen gebonden variabelen gemeen hebben, en dus kunnen we Stelling 13.6 *ii)* gebruiken om alle kwantoren uit φ_1 naar voren te halen:

$$\varphi \equiv Q_1^1 x_1 \ldots Q_n^1 x_n\, (\psi_1 \bullet \varphi_2)$$

Vervolgens gebruiken we Stelling 13.6 *iii)* en compositionaliteit om te besluiten tot

$$\varphi \equiv Q_1^1 x_1 \ldots Q_n^1 x_n\, Q_1^2 y_1 \ldots Q_m^2 y_m\, (\psi_1 \bullet \psi_2)$$

waarvan het rechterlid in prenexvorm is.

iv) Stel $\varphi = \varphi_1 \to \varphi_2$. Met behulp van Stelling 13.6 *iv)* en *v)* vinden we als prenex-vorm:

$$\varphi \equiv (Q_1^1)'x_1 \ldots (Q_n^1)'x_n\, Q_1^2 y_1 \ldots Q_m^2 y_m\, (\psi_1 \to \psi_2)$$

v) Stel $\varphi = Qx\varphi_1$. Dan geldt $\varphi \equiv Qx\alpha_1$, en het rechterlid is in prenex-vorm.

We kunnen in feite uit het bewijs van Stelling 13.7 ook een algoritme destilleren om een gegeven formule φ te herschrijven tot een equivalente formule φ' die in prenex-vorm is. Het volgende voorbeeld illustreert één en ander.

Voorbeeld 13.2 Bekijk

$$\varphi = \forall x(Rxy \to \exists z \neg\exists x Rxz) \wedge \forall z(\forall n T(n,x,z) \to \forall z \exists u \exists u P(x,u,u)).$$

Uit Voorbeeld 13.1 weten we dat we er de volgende equivalente eenduidige formule voor kunnen vinden:

$$\varphi \equiv \forall v(Rvy \to \exists s \neg\exists q Rqs) \wedge \forall z(\forall n T(n,x,z) \to \forall w \exists r \exists u P(x,u,u)).$$

We verwijzen steeds naar de stappen *i)-v)* uit Stelling 13.7. Bekijken we eerst de deelformule $\varphi_1 = \forall v(Rvy \to \exists s \neg\exists q Rqs)$. Met stap *i)* vinden we

$$\varphi_1 = \forall v(Rvy \to \exists s \forall q \neg Rqs)$$

en met stap *v)* vervolgens

$$\varphi_1 = \forall v \exists s \forall q(Rvy \to \neg Rqs)$$

Nu deelformule $\forall z(\forall n T(n,x,z) \to \forall z \exists u \exists u P(x,u,u))$. Met stap *v)* vinden we

$$\varphi_2 = \forall z \forall w \exists r \exists u(\forall n T(n,x,z) \to P(x,u,u))$$

en dan met stap *iv)*:

$$\varphi_2 = \forall z \forall w \exists r \exists u \exists n(T(n,x,z) \to P(x,u,u))$$

We vinden dus voor φ:

$$\varphi \equiv \forall v \exists s \forall q(Rvy \to \neg Rqs) \wedge \forall z \forall w \exists r \exists u \exists n(T(n,x,z) \to P(x,u,u))$$

Hier passen we tot slot stap *ii)* en *iii)* toe om te concluderen:

$$\forall v \exists s \forall q \forall z \forall w \exists r \exists u \exists n((Rvy \to \neg Rqs) \wedge (T(n,x,z) \to P(x,u,u))).$$

Opgave 13.9 Geef prenex-normaalvormen voor de formules uit Opgave 13.6.

13.3 Skolem normaalvorm

Voorbeeld 13.3 Bekijk de volgende zin:

Voor iedereen is er wel iemand eerder geboren. (9)

Met als vertaalsleutel Exy voor "x is eerder geboren dan y", stellen we voor

$$\forall x \exists y Eyx \tag{10}$$

(De rol van x en$y$ is omgedraaid in de laatste formule.)

Als we nadenken over de waarheid van deze formule op het domein van alle mensen, dan zou een redenering kunnen zijn: iedereen heeft een vader, en de vader van iemand is altijd ouder dan die iemand, dus de bewering klopt. We zouden de functie *vader_van(x)* kunnen weergeven door $f(x)$, en dan (10) vertalen door:

$$\forall x, Ef(x)x \tag{11}$$

De variabele y (waarvan we weten dat deze afhangt van x) is dus verdwenen, en vervangen door een functieterm die zichtbaar afhangt van x.

Voorbeeld 13.4 Bekijk de volgende zin:

Voor elk tweetal verschillende getallen is er een derde dat er tussenin ligt. (12)

Met een voor de hand liggende vertaalsleutel vinden we:

$$\forall x \forall y \exists z(x < y \to (x < z \wedge z < y)) \tag{13}$$

Dat er zo'n derde bij elk tweetal kan worden gevonden kan worden opgevat als het resultaat van een binaire functie $x, y \mapsto g(x,y)$. Hoewel g niet expliciet gedefinieerd is, zou dat in principe kunnen, als (12) waar is. Dat geeft

$$\forall x \forall y(x < y \to (x < g(x,y) \wedge g(x,y) < y)) \tag{14}$$

De functies f en g heten ook wel *Skolem functies*, en het proces dat vanuit (13) de formule (14) oplevert heet ook wel *skolemiseren*. Merk op dat het resultaat een universele formule is: een formule met alleen $\forall$-kwantoren.

Definitie 13.9

$$\varphi = \forall x_1 \ldots \forall x_n \exists y \psi$$

een formule in de eerste orde taal FOR(L)*, en f een functiesymbool dat niet in L voorkomt. Beschouw nu de volgende formule*

$$\varphi' = \forall x_1 \ldots \forall x_n [f(x_1, \ldots, x_n)/y]\psi$$

Dan is φ' een formule in de taal $L' \supseteq L$. We noemen f een skolemfunctie*, en we zeggen dat we de existentiële kwantor y hebben* geskolemiseerd*.*

Er is het volgende verband tussen een formule φ en zijn skolemisatie φ':

Stelling 13.8 (Skolemisatie) *Laat φ' een skolemisatie zijn van φ. Dan*

$$\varphi \textit{ is vervulbaar} \Leftrightarrow \varphi' \textit{ is vervulbaar.}$$

Bewijs: Opgave.

Opgave 13.10 Bewijs Stelling 13.8

Twee formules die tegelijkertijd vervulbaar of onvervulbaar zijn worden *vervulbaarheidsequivalent* genoemd. Dit wordt soms geschreven als $\varphi \equiv_{SAT} \varphi'$. (De afkorting SAT staat voor *satisfiability*.)

Let op: een Skolemisatie is in het algemeen *niet* logisch equivalent met de oorspronkelijke formule:

Voorbeeld 13.5 Een Skolemisatie van

$$\forall x \exists y (x < y) \tag{15}$$

is

$$\forall x (x < f(x)). \tag{16}$$

(We hadden voor het functiesymbool f ook een andere functiesymbool kunnen kiezen.)

Dankzij de Skolemisatiestelling (Stelling 13.8) weten we nu dat (15) en (16) vervulbaarheidsequivalent zijn. Ze zijn beiden dus vervulbaar, of beiden onvervulbaar. Makkelijk is in te zien dat beiden vervulbaar zijn: neem $\mathscr{M} = \langle \mathbb{N}, <, f \rangle$, met $I(f)$ de successor-functie:

$$I(f) : \mathbb{N} \to \mathbb{N} : x \mapsto x + 1,$$

en $I(<)$ de kleiner-dan relatie op $\mathbb{N}$:

$$(m,n) \in I(<) \quad \Leftrightarrow \quad I(m) \text{ is strict kleiner dan } I(n).$$

Dan geldt (zelfs voor iedere bedeling b) dat

$$\mathscr{M}, b \models \forall x \exists y (x < y) \text{ en } \mathscr{M}, b \models \forall x (x < f(x)).$$

Maar als f anders wordt geïnterpreteerd, en er dus een ander model, $\mathscr{M}'$, wordt genomen, bijvoorbeeld via

$$I'(f) : \mathbb{N} \to \mathbb{N} : x \mapsto x - 1,$$

dan $\mathscr{M}, b \models \forall x \exists y (x < y)$ maar $\mathscr{M}', b \not\models \forall x \exists y (x < f(x))$. Vanwege dit feit zijn (15) en (16) niet logisch equivalent. M' kiest als het ware een verkeerde interpretatie voor f, zodat de betekenis van (15) en (16) uiteen gaan lopen. Einde voorbeeld.

Vervulbaarheidsequivalentie is enorm belangrijk voor stellingenbewijzers. Met stellingenbewijzers wordt de geldigheid van $\Gamma \models_? \varphi$ namelijk onderzocht door na te gaan of $\Gamma \cup \{\neg\varphi\}$ vervulbaar is. (Herinner dat een negatief antwoord voor de ene expressie een positief antwoord voor de andere is.) Het gaat bij stellingenbewijzers dus om *vervulbaarheid* en vervulbaarheidsequivalentie, en niet om het (veel sterkere) logische equivalentie.

Dat laatste komt goed uit, want hoewel hoogstwaarschijnlijk[2] geen efficiënte methoden bestaan om CNFs te berekenen die logisch equivalent zijn aan de originele formule (dit werd uitgelegd aan het einde van Hoofdstuk 8 over normaalvormen), bestaan er wél efficiënte methoden CNFs te berekenen die SAT-equivalent zijn aan de originele formule. Eén zo'n methode werkt middels het berekenen van de zg. Tseitin-afgeleide. Een uitleg daarvan is niet moeilijk maar valt buiten het bestek van deze tekst.

Stelling 13.9 *Voor elke formule φ is er een formule φ_S, ook wel genoemd de skolemnormaalvorm van φ, met de eigenschappen*

- *De formule φ_S is vervulbaarheidsequivalent met φ.*
- *De formule φ_S is een* universele formule*, i.e., van de vorm $\forall x_1 \ldots \forall x_k \psi$, $k \geq 0$, met ψ kwantorvrij. We noemen dit ook wel een skolemvorm.*

Bewijs: We brengen φ eerst in prenexnormaalvorm, we vinden dan een formule

$$Q_1 x_1 \ldots Q_n x_n \psi, \text{ met } \psi \text{ kwantorvrij.}$$

Zoek nu het eerste voorkomen van een existentiële kwantor in de matrix $Q_1 x_1 \ldots Q_n x_n$, laten we zeggen dat dat $Q_i x_i$ is. Met Definitie 13.9 kunnen we deze kwantor skolemiseren:

$$\forall x_1 \ldots \forall x_{i-1} Q_{i+1} x_{i+1} \ldots Q_n x_n [f(x_1, \ldots, x_{i-1})/x_i] \psi$$

Dit proces kan worden voortgezet totdat alle existentiële kwantoren zijn verdwenen, waarbij het resultaat steeds vervulbaar is precies dan als ψ vervulbaar is. Uiteindelijk, als er geen existentiële kwantoren meer in de matrix zitten, vinden we φ'.

Opgave 13.11 Skolemiseer de volgende formules.

1. $\forall x \exists y Q x y$
2. $\exists x \exists y Q x y$
3. $\exists x \exists y \forall z (Q x y \wedge R y z)$

[2] Tenzij P = NP.

4. $\forall z \exists x \exists y (Qxy \wedge Ryz)$
5. $\forall x \forall y \exists z \exists u (Rxyz \rightarrow (Pyzu \rightarrow Qf(uz)))$
6. $\forall x \forall y \exists z \exists u (Ruz \rightarrow (Pz \rightarrow Qf(u)))$
7. $\neg \exists x \exists y (Rxy \rightarrow \forall u (Tyu))$
8. $\exists x \exists y \forall z (Rxy \rightarrow Qzx)$
9. $\forall y \exists x Rxy \rightarrow \exists z \forall x Qxz$
10. $\exists x \forall y Rxy \rightarrow \forall x \exists z Qxz$

Hoofdstuk 14

Bewijssystemen voor de predikatenlogica

Het systeem van natuurlijke deductie, dat we in Hoofdstuk 9 invoerden voor de propositielogica, kan worden uitgebreid tot de predikatenlogica. Dat wil zeggen dat de regels voor de propositionele connectieven van kracht blijven en dat er nieuwe introductie- en gebruiksregels worden ingevoerd voor de kwantoren. We zullen in deze paragraaf de introductie- en gebruiksregels voor de nieuwe taal-elementen bespreken. We doen dat alleen voor de logische symbolen, in het bijzonder voor de kwantoren $\forall$ en $\exists$, en voor het speciale predikaatsymbool "=". De logische symbolen die dan niet aan de orde komen zijn de hulpsymbolen, dus de haakjes (als we heel precies de definitie van formules beschouwen, zijn alle introductieregels—en eventueel de $\bot$-regel—regels die haakjes introduceren, net zoals de gebruiksregels opgevat kunnen worden als manieren om haakjes kwijt te raken), en de komma (die slechts dienst doet om argumenten in functies en predikaten te scheiden).

Ons basissysteem zal geen regels geven voor de taal-afhankelijke functie- en predikaatsymbolen: de logica wil zich uiteraard niet vastleggen op eigenschappen van algemene predikaten $P, Q, R, P_0, P_1, \ldots$ en functies $F, G, H, F_0, F_1, \ldots$. Wel zouden (wiskundige) theorieën extra regels voor deze symbolen kunnen toevoegen; een systeem voor de verzamelingenleer zou eigenschappen van de het predikaat "is deelverzameling van", en de functies 0-plaatsige functie "$\emptyset$, de 1-plaatsige functie ^{C} ("complement van"), en de 2-plaatsige functie $\cup$ ("de vereniging van") kunnen vastleggen, en een systeem voor de meetkunde predikaten als "is een lijn", "is een punt", "lopen evenwijdig" en "ligt op". Merk op dat, hoewel de logica dus geen introductieregels geeft voor deze niet-logische symbolen, we er toch wel algemene eigenschappen over willen kunnen afleiden, zoals bijvoorbeeld $\forall x((x \text{ is een lijn}) \vee \neg(x \text{ is een lijn})$.

Voordat we nu de regels voor de predikatenlogica gaan bespreken, herinneren we ons even dat we een calculus proberen te geven waarmee we beweringen van het type

$$\Gamma \vdash \varphi$$

willen bewijzen, waarbij we Γ weer een verzameling premissen noemen en φ de conclusie. Voor de propositionele connectieven mogen we de oude regels uit Hoofdstuk 9 gebruiken. Merk nog op dat, als we nu algemene logische redeneringen willen proberen te modelleren in onze calculus, we eerst nog een (vaak niet triviale) vertaalslag moeten maken. De volgende opgave geeft daarvan een voorbeeld.

Opgave 14.1 We vragen ons af of de volgende gevolgtrekking logisch juist is. Hierin hebben we te maken met de premissen $P1$ en $P2$, en een conclusie C:

P1 : Iedereen houdt van Elvis
P2 : Elvis houdt alleen van mij
C : Ik ben Elvis

1. Vertaal de zinnen $P1$, $P2$ en C in de predikaatlogische taal; kies geschikte constanten en predikaten. Noem de vertalingen respectievelijk $p1$, $p2$ en c.
2. Is het wenselijk dat $p1, p2 \vdash c$? Wat is een goed criterium voor zo'n vraag?

14.1 Natuurlijke deductie

De regels voor gelijkheid liggen zo voor de hand, dat we ze meteen geven:

$$\frac{}{t = t}\ \text{I=} \qquad \frac{s = t,\ [s/x]\varphi}{[t/x]\varphi}\ \text{G=}$$

De introductieregel zegt dat het per definitie bewijsbaar is dat elke term gelijk is aan zichzelf, en de gebruiksregel dat, als eenmaal de gelijkheid van twee termen bewezen is, we deze voor elkaar mogen substitueren in eenmaal bewezen formules. Hoewel deze regels overduidelijk correct zijn voor het predikaat "gelijk zijn", is het feit dat ze ook voldoende (dus volledig) voor dit predikaat zijn, misschien alvast vermelding waard.

Opgave 14.2 Laat zien dat "=" bewijsbaar een equivalentierelatie is, d.w.z. bewijs dat:

1. $\vdash t = t$
2. $s = t \vdash t = s$
3. $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$

We zullen nu stilstaan bij de regels voor de kwantoren; we zullen voorbeelden geven van afleidingen en tevens laten zien wat er mis kan gaan als de restricties overschreden worden. We doen dit per regel, die we steeds eerst geven, met de bijbehorende restricties.

Bij de kwantorregels spelen steeds drie zaken een rol: de *formule* waarop de regel wordt toegepast (in de definitie van de regels is dit steeds een φ) een *variabele* die relevant is (in de definitie steeds een x) en een *term* waarmee iets gebeurt (in de definitie een constante-symbool a of een algemeen term-symbool t). Om

bewijzen iets leesbaarder te maken, zullen we daarom af en toe in een bewijs, op de plaats waar een kwantorstap plaats vindt, ook het drietal [for, var, ter] expliciet specificeren. Dit is niet een onderdeel van het bewijs zelf (we zullen dit achter een verticale lijn doen) en moet in principe altijd uit het bewijs zelf te halen zijn.

De existentiële kwantor. Allereerst de introductieregel voor de existentiële kwantor $\exists$. Er is een bewijs voor $\exists x\varphi$, als we φ kunnen bewijzen voor een of andere term t:

$$\mathrm{I}\exists\ \frac{[t/x]\varphi}{\exists x\varphi},\ t \text{ vrij voor } x \text{ in } \varphi$$

De term t moet dus vrij zijn voor x in φ. We laten zien wat er mis kan gaan als deze restrictie niet in acht wordt genomen, door een foute afleiding te geven van $\forall y(Ryy)$ naar $\exists x\forall y(Ryx)$.

Opgave 14.3

Laat zien dat $\forall y(Ryy) \not\models \exists x\forall y(Ryx)$.

We geven nu een fout bewijs voor $\exists x\forall y(Ryx)$ uit $\forall y(Ryy)$; een niet correcte toepassing van de I$\exists$-regel wordt daarbij aangegeven met I$\exists\!\!\not$.

1.	$\forall y(Ryy)$	ass
2.	$\exists x\forall y(Ryx)$	I$\exists\!\!\not$, 1

Dit bewijs is fout omdat y niet vrij is voor x in $\forall y(Ryx)$.

Nu een voorbeeld van een correct gebruik van I$\exists\!\!\not$. We bewijzen $Rab \vdash \exists x\exists y Rxy$.

1.	Rab	ass
2.	$\exists y\, Ray$	I$\exists$, 1
3.	$\exists x\exists y\, Rxy$	I$\exists$, 2

In woorden hebben we nu bewezen dat, als de relatie R geldt tussen twee objecten a en b, er objecten zijn waartussen de relatie R geldt.

Met één bewijsregel kunnen nu al verschillende existentiële formules afgeleid worden:

Opgave 14.4 1. Bewijs de volgende beweringen. Wat is de sterkste van de vier conclusies, wat de zwakste?

(a) $Raa \vdash \exists x\, Rxx$

(b) $Raa \vdash \exists x\exists y\, Rxy$

(c) $Raa \vdash \exists x\, Rax$

(d) $Raa \vdash \exists x\, Raa$

2. Laat zien dat $Rab \not\models \exists x\, xRxx$. Laat zien dat het bewijs van (1a) ook niet aangepast kan worden om te bewijzen dat $Rab \vdash \exists x\, xRxx$.

De gebruiksregel voor $\exists$. Als we $\exists x\varphi$ eenmaal hebben bewezen, dan kunnen we laten zien dat ψ volgt, als we kunnen laten zien dat ψ volgt uit een willekeurige instantie van φ (zolang ψ maar niet over die specifieke instantie gaat).

$$\mathrm{G}\exists\ \frac{\exists x\varphi,\ [a/x](\varphi \to \psi)}{\psi},\ a \text{ niet in open aanname, } \varphi \text{ of } \psi$$

We laten zien, waarom de restrictie "a niet in open aanname" nodig is, door een foute afleiding te geven voor

$$a \neq b \to b = c \wedge b \neq c.$$

Overigens, gelijkheid en ongelijkheid binden heel sterk, en conjunctie bindt sterker dan implicatie. Parseer dus als $(a \neq b) \to ((b = c) \wedge (b \neq c))$.

Opgave 14.5 Laat zien dat $\not\models a \neq b \to b = c \wedge b \neq c$.

Hier volgt de beloofde pseudo-afleiding.

1.	$a \neq b$	ass
2.	$b = b$	I=
3.	$\exists x\, x = b$	I$\exists$, 2
4.	$a = b$	ass
5.	$\bot$	G$\neg$, 1, 4
6.	$a = c \wedge a \neq c$	$\bot$-regel, 5
7.	$a = b \to b = c \wedge b \neq c$	I$\to$, 4–6
8.	$b = c \wedge b \neq c$	I$\exists\!\!\not$, 3, 7
9.	$a \neq b \to b = c \wedge b \neq c$	I$\to$, 1–8

Opgave 14.6 1. Laat zien dat

$$\exists x\, x \neq a \ \not\models\ b = c \wedge b \neq c. \tag{1}$$

2. Laat zien dat zonder de restrictie "a niet in φ" de identiteit (1) afgeleid kan worden.

3. Laat zien dat

$$\exists x Rxb \not\models Rab. \tag{2}$$

4. Laat zien dat zonder de restrictie "a niet in ψ" de identiteit (2) afgeleid kan worden.

We laten nu het correct gebruik van G$\exists$ zien in een afleiding van $\exists x\exists y Rxy \vdash \exists x\exists y Ryx$:

1.	$\exists x\exists y Rxy$	ass
2.	$\exists y Ray$	ass
3.	Rab	ass
4.	$\exists y Ryb$	I$\exists$, 3
5.	$\exists x\exists y Ryx$	I$\exists$, 4
6.	$Rab \to \exists x\exists y Ryx$	I$\to$, 3–5
7.	$\exists x\exists y Ryx$	G$\exists$, 2, 6
8.	$\exists y Ray \to \exists x\exists y Ryx$	I$\to$, 2–7
9.	$\exists x\exists y Ryx$	G$\exists$, 1, 8

Opgave 14.7 Geef afleidingen voor:

1. $\exists x Rxx \vdash \exists x \exists y Rxy$

2. $\exists x(Ax \wedge Bx) \vdash \exists x Ax \wedge \exists x Bx$

De universele kwantor. De introductieregel voor $\forall$. We hebben een bewijs voor $\forall x\varphi$ als we φ kunnen bewijzen voor een willekeurige term a op de plek van x.

$$\text{I}\forall: \quad \frac{[a/x]\varphi}{\forall x\varphi},\ a \text{ niet in open aanname, of in } \varphi$$

We laten zien waarom de restrictie "a niet in φ" nodig is, door een fout bewijs voor $\forall x(x = a)$ geven:

1.	$a = a$	I=
2.	$\forall x(x = a)$	I$\forall\!\!\not{}$, 2

Opgave 14.8 1. Laat zien dat $\not\models \forall x(x = a)$.

2. Laat zien dat

$$\not\models a \neq b \to \forall x(x \neq x) \tag{3}$$

3. Laat zien dat zonder de restrictie "a niet in open aanname" de identiteit (3) afgeleid kan worden.

De gebruiksregel voor $\forall$. Als we een bewijs hebben voor $\forall x\varphi$, hebben we een bewijs voor φ met een willekeurige term t voor x in φ:

$$\text{G}\forall: \quad \frac{\forall x\varphi}{[t/x]\varphi},\ t \text{ vrij voor } x \text{ in } \varphi$$

Zonder deze restrictie zou $\exists y(y \neq y)$ afgeleid kunnen worden uit $\forall x \exists y(x \neq y)$:

1.	$\forall x \exists y(x \neq y)$	ass
2.	$\exists y(y \neq y)$	G$\forall\!\!\not{}$, 1

Opgave 14.9 Geef een bewijs van de volgende identiteiten.

1. $\forall x Rxx \vdash Raa$

2. $\forall x \forall y Rxy \vdash Rab \wedge Rcc$

We geven nog één voorbeeld van een afleiding waarbij alle kwantor-regels een rol spelen, namelijk het bewijs van $\exists x \forall y Rxy \vdash \forall y \exists x Rxy$:

1.	$\exists x \forall y Rxy$	ass
2.	$\quad \forall y Ray$	ass
3.	$\quad Rab$	G$\forall$, 2
4.	$\quad \exists x Rxb$	I$\exists$, 3
5.	$\forall y Ray \to \exists x Rxb$	I$\to$, 2–4
6.	$\exists x Rxb$	G$\exists$, 1, 5
7.	$\forall y \exists x Rxy$	I$\forall$, 6

Overzicht van alle predikaat-regels in Fitch

I= :	$\dfrac{}{t = t}$	
G= :	$\dfrac{s = t,\ [s/x]\varphi}{[t/x]\varphi}$	
I$\forall$:	$\dfrac{[a/x]\varphi}{\forall x\varphi}$,	– a niet in open aanname – a niet in φ
G$\forall$:	$\dfrac{\forall x\varphi}{[t/x]\varphi}$,	t vrij voor x in φ
I$\exists$:	$\dfrac{[t/x]\varphi}{\exists x\varphi}$,	t vrij voor x in φ
G$\exists$:	$\dfrac{\exists x\varphi,\ [a/x](\varphi \to \psi)}{\psi}$,	– a niet in open aanname – a niet in φ – a niet in ψ

Merk op dat de kwantorenregels zó zijn geplaatst dat de premissen en conclusies in elkaar overlopen.

Opgave 14.10 1. Laat zien dat:

(a) $\forall x(Px \wedge Qx) \vdash \forall x Px \wedge \forall x Qx$

(b) $\forall x Px \wedge \forall x Qx \vdash \forall x(Px \wedge Qx)$

(c) $\forall x \forall y Rxy \vdash \forall x \forall y(Rxy \wedge Ryx)$

(d) $\forall x(Px \to Qx),\ \forall x Px \vdash \forall x Qx$

(e) $\neg \exists x Px \vdash \forall x \neg Px$ (Hint: probeer $\neg Pa$ als tussenresultaat af te leiden.)

(f) $\neg \exists x \neg Px \vdash \forall x Px$ (Hint: probeer $\neg\neg Pa$.)

(g) $\exists x \exists y Pxy \vdash \exists y \exists x Pxy$

(h) $\exists y \forall x Pxy \vdash \forall x \exists y Pxy$

(i) $\neg \forall x \neg Px \vdash \exists x Px$

2. Laat zien dat:

(a) $\exists x(Px \wedge Qx) \vdash \exists x Px \wedge \exists x Qx$

(b) $\forall x(Px \to Qx),\ \exists x Px \vdash \exists x Qx$

(c) $\exists x \neg Px \vdash \neg \forall x Px$

(d) $\forall x \neg Px \vdash \neg \exists x Px$

(e) $\neg \forall x Px \vdash \exists \neg Px$

(f) $\forall x(Px \to Qx),\ \exists x \neg Qx \vdash \exists x \neg Px$

(g) $\forall x(Px \vee Qx),\ \exists x \neg Qx \vdash \exists x Px$

(h) $\forall x(Px \to Qx),\ \exists x(Px \wedge Rx) \vdash \exists x(Qx \wedge Rx)$

Opgave 14.11 Bekijk nog eens Opdracht 14.1. Bewijs daarna formeel dat $p1, p2 \vdash c$.

14.2 Hilbert's systeem

We definiëren weer een axioma-verzameling. Eerst nemen we de axioma-schema's van de propositielogica over:

Axioma 14.1 $\varphi \to (\psi \to \varphi)$.

Axioma 14.2 $(\varphi \to (\psi \to \chi) \to ((\varphi \to \psi) \to (\varphi \to \chi))$.

Axioma 14.3 $(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$.

Er is echter meer, want ook de kwantoren doen nu mee. We geven eerst een voorlopige formulering van een vierde axioma-schema:

Axioma 14.4 (voorlopige versie:)
$\forall v\varphi \to [t/v]\varphi$ *voor elke individuele variabele* v *en elke term* t.

Dit schema drukt uit dat een universele bewering elk van zijn 'toepassingen op een of ander bijzonder geval' impliceert. Dus: $\forall xPx \to Pa$ is een axioma van deze vorm. Er is echter een moeilijkheid verbonden aan de voorlopige versie van dit laatste axioma-schema.

Opgave 14.12 Waarom kan de formule

$$\forall y\neg\forall x\ x = y \to [x/y]\neg\forall x\ x = y,$$

die voldoet aan de voorlopige versie van axioma-schema 4, geen axioma zijn? Met andere woorden: wat gaat er mis wanneer we in dit geval x voor y substitueren in de deelformule $\neg\forall x\ x = y$?

Uit het probleem in Opgave 14.12 blijkt dat de voorlopige versie van 4. te ruim is geformuleerd. We moeten een beperking opleggen aan het soort termen t dat voor v mag worden gesubstitueerd in φ. In wat nu volgt gaan we voor het gemak uit van een predikatenlogische taal die geen functiesymbolen bevat. Met *term* bedoelen we dus: constante of variabele. Hoe de functiesymbolen zouden moeten worden verdisconteerd moet je misschien zelf even bedenken. We voeren het volgende begrip in teneinde de beperking die we aan 4. willen opleggen te formuleren.

Definitie 14.1 *De term* t *is* **vrij substitueerbaar** *voor de variabele* v *in de formule* φ *wanneer* t *ofwel een constante is, ofwel een variabele die vrij voorkomt in* $[t/v]\varphi$ *op elke plaats waar* v *vrij voorkwam in* φ.

Gauw een paar voorbeelden. a is vrij substitueerbaar voor y in $\forall xRxy$, want a is een constante. z is vrij substitueerbaar voor y in $\forall xRxy$, want $[z/y]\forall xRxy$ is gelijk aan $\forall xRxz$, en hierin is z vrij. x is *niet* vrij substitueerbaar voor y in $\forall xRxy$, want $[x/y]\forall xRxy$ is gelijk aan $\forall xRxx$, en hierin is x *niet* vrij op de plaats waar y eerst *wel* vrij was: de variabele wordt als het ware 'ingevangen' door de universele kwantor. z is *niet* vrij substitueerbaar voor y in $\forall z(Pz \to \exists xRxy)$; probeer zelf na te gaan waarom. Nog wat jargon: in plaats van *vrij substitueerbaar voor* zegt men ook wel *vrij voor* of *substitueerbaar voor*.

Opgave 14.13 Ga na of x vrij substitueerbaar is voor y in de volgende formules:

1. $Py \wedge \exists xRxy$.
2. $Px \wedge \exists zRzy$.
3. $\forall y(Pz \to \exists xRxy)$.
4. $\forall x(Py \to \exists yRxy)$.
5. $\exists x(Px \wedge \exists yRxy)$.
6. $Py \wedge \forall x\exists yRxy$.

Nu we de beschikking hebben over het begrip *vrij substitueerbaar zijn voor* kunnen we de correcte versie van axioma-schema 4. formuleren:

Axioma 14.4 $\forall v\varphi \to [t/v]\varphi$ *voor elke individuele variabele* v *en elke term* t *die vrij substitueerbaar is voor* v *in* φ.

Het nu volgende axioma-schema 5. is een beetje flauw. Het is nodig omdat we nu eenmaal hebben besloten om loze kwantificatie toe te staan.

Axioma 14.5 $\varphi \to \forall v\varphi$ *mits* v *niet vrij voorkomt in* φ.

Tenslotte hebben we:

Axioma 14.6 $\forall v(\varphi \to \psi) \to (\forall v\varphi \to \forall v\psi)$.

Dat waren de axioma-schema's. De verhouding van de echte axioma's tot de schema's is hier iets ingewikkelder dan in het geval van de propositielogica. Om de verzameling van predikatenlogische axioma's te kunnen definiëren moeten we eerst definiëren wat een generalisering van een formule is:

Definitie 14.2 *Een* **generalisering** *van een formule* φ *is een formule die ontstaat door 0 of meer universele kwantoren (met bijbehorende variabelen) te schrijven voor* φ.

Dus: Px, $\forall xPx$, $\forall x\forall zPx$,... zijn generaliseringen van Px. De verzameling predikatenlogische axioma's is nu als volgt gedefinieerd:

Definitie 14.3 *Een* **predikatenlogisch axioma** *is een generalisering van een formule volgens een van de schema's 1. tot en met 6.*

Voorbeelden van axioma's zijn:

- $\forall x(Px \to (Qx \to Px))$ [een generalisering van 1];
- $\forall xPx \to Pa$ [een generalisering van 4];
- $\forall x(\forall yPy \to Px)$ [een generalisering van 4];
- $\forall x(Px \to \forall yPx)$ [een generalisering van 5];
- $\forall x(Px \to Qx) \to (\forall xPx \to \forall xQx)$ [een generalisering van 6];
- $\forall z\forall y(\forall x(Px \to Qx) \to (\forall xPx \to \forall xQx))$ [een generalisering van 6].

Wanneer we een axiomaverzameling willen definiëren voor een predikatenlogische taal waarin ook het identiteitsteken wordt gebruikt hebben we nog twee extra axioma's nodig:

Axioma 14.7 $v = v$ *voor elke variabele* v.

Axioma 14.8
$(v_1 = w_1 \to (\ldots(v_n = w_n \to (Av_1 \ldots v_n \to Aw_1 \ldots w_n))\ldots))$ *voor elke n-plaatsige predikaatletter A*
en voor alle variabelen $v_1,\ldots,v_n$, $w_1,\ldots,w_n$.

Axioma-schema 8. ziet er misschien een beetje ingewikkeld uit, maar het idee is gewoon: in atomaire formules moet je variabelen die naar hetzelfde ding verwijzen voor elkaar kunnen substitueren. Een andere formulering van 8. is deze:

Axioma 14.8 (herformulering:)
$(v_1 = w_1 \wedge \ldots \wedge v_n = w_n \wedge Av_1 \ldots v_n) \to Aw_1 \ldots w_n$.

De 'implicatie-versie' die wij hebben gekozen verdient echter de voorkeur omdat de afleidingsregel Modus Ponens er mooier bij aansluit.

Wanneer we een calculus willen geven voor een taal die behalve het identiteitsteken ook functiesymbolen bevat hebben we nog een negende axioma nodig, om te garanderen dat we in termen die van de vorm $gt_1 \ldots t_n$ zijn, termen voor $t_1,\ldots,t_n$ mogen substitueren mits ze naar hetzelfde individu verwijzen. We laten dat axioma hier nu maar voor het gemak achterwege.

Hier volgen enkele voorbeelden van generaliseringen van 7. en 8.

- $\forall x\, x = x$
- $\forall x \forall y (x = y \to (Px \to Py))$
- $(x = y \to (Px \to Py))$
- $\forall x \forall y \forall z \forall u (x = y \to (z = u \to (Rxz \to Ryu)))$.

De definitie van de verzameling axioma's van een predikatenlogische taal met identiteit wordt nu:

Definitie 14.4 *Een* **predikatenlogisch axioma** *voor een predikatenlogische taal* $\mathcal{T}$ *met identiteit is een generalisering van een formule volgens een van de axioma-schema's 1. tot en met 8.*

De afleidingsregel is weer *Modus Ponens*: concludeer uit φ en $\varphi \to \psi$ tot ψ.

Tenslotte voeren we $\wedge$, $\vee$, $\leftrightarrow$ and $\exists$ als afkortingen in. Naast de definities uit § 9.6 hebben we nog nodig:

- $\exists v \varphi$ is een afkorting voor $\neg \forall v \neg \varphi$.

Hiermee hebben we een deductief systeem (of: een axiomatiek, of: een calculus) voor de predikatenlogica gegeven. Er bestaan vele andere deductieve systemen voor de predikatenlogica die *equivalent* zijn in de zin dat ze dezelfde verzameling formules als stellingen opleveren. De definities van *afleiding* (of: *bewijs*), en *stelling* zijn als bij de propositielogica.

Net als bij de propositie-logische calculus die we in § 9.5 gepresenteerd hebben, moeten we weer onderscheid maken tussen bewijzen *in* de calculus en bewijzen *over* de calculus. Hier is een voorbeeld van een bewijs *in* de calculus.

Stelling 14.1 *Voor elke constante* a *is de formule* $a = a$ *een stelling.*

Bewijs:

1.	$\forall x\, x = x$	[axioma volgens schema 7]
2.	$\forall x\, x = x \to a = a$	[axioma volgens schema 4]
3.	$a = a$	[MP uit 1 en 2].

□

De notatie voor "$a = a$ is een stelling" is weer: $\vdash a = a$. We roepen de notatie in herinnering voor "φ is afleidbaar uit formuleverzameling Γ":

$$\Gamma \vdash \varphi.$$

Voorbeeld: laat Γ de verzameling $\{\forall x Rxx\}$ zijn. Dan kunnen we een afleiding geven van Raa:

Stelling 14.2 $\forall x Rxx \vdash Raa$.

Bewijs:

1.	$\forall x Rxx$	[volgens aanname]
2.	$\forall x Rxx \to Raa$	[axioma volgens schema 4]
3.	Raa	[MP uit 1 en 2].

□

Een andere notatie die ook wel wordt gebruikt voor het resultaat uit de stelling is: $\forall x Rxx \vdash Raa$.

Opgave 14.14 Laat zien:

$$\forall x(Ax \to Bx), Aa \vdash Ba.$$

Opgave 14.15 Laat zien:

$$\forall x(\neg Ax \to \neg Bx), \forall x(Ax \to Cx), Ba \vdash Ca.$$

Hier is nog een voorbeeld van een bewijs van een stelling van de calculus.

Stelling 14.3 $\vdash x = y \to y = x$.

Bewijs:

1.	$x = y \to (x = x \to (x = x \to y = x))$	[ax 8]
	NB = is een tweeplaatsig predikaat	
2.	$(x = y \to (x = x \to (x = x \to y = x))) \to$ $((x = y \to x = x) \to$ $(x = y \to (x = x \to y = x)))$	[ax 2]
3.	$(x = y \to x = x) \to (x = y \to (x = x \to y = x))$	[MP uit 1,2]
4.	$x = x$	[ax 7]
5.	$x = x \to (x = y \to x = x)$	[ax 1]
6.	$x = y \to x = x$	[MP uit 4,5]
7.	$x = y \to (x = x \to y = x)$	[MP uit 3,6]
8.	$(x = y \to (x = x \to y = x)) \to$ $((x = y \to x = x) \to (x = y \to y = x))$	[ax 2]
9.	$(x = y \to x = x) \to (x = y \to y = x)$	[MP uit 7,8]
10.	$x = y \to y = x$	[MP uit 6,9].

□

Je ziet het: een simpel principe als $x = y \to y = x$ vereist al het één en ander aan formule-manipulatie. Overigens zijn, afgezien van het gebruik van de axioma's 7 en 8, alle stappen in het bewijs puur propositielogische stappen; kwantormanipulatie speelt in het bewijs geen rol.

Het is niet de bedoeling dat je je het hoofd breekt over het vinden van dit soort bewijzen. Wel is het nuttig dat je een idee heeft van wat een *bewijs in de predikaatlogische calculus* is. Vandaar de bovenstaande voorbeelden.

Weer is—net als in het propositielogische geval—het onderscheid tussen een bewijs *in* en een bewijs *over* de calculus van groot belang. Hier is een voorbeeld van een metastelling voor de predikatenlogische calculus:

Stelling 14.4 *De predikatenlogische calculus voldoet aan het principe van Universele Generalisatie: Als* $\vdash \varphi$ *dan ook* $\vdash \forall v\varphi$.

Bewijs: We gebruiken weer inductie naar de lengte van een bewijs in de calculus.

- *Basisstap*: het bewijs van φ bestaat uit het rijtje $\langle\varphi\rangle$ dat alleen de formule φ bevat. In dit geval moet φ een axioma zijn. Maar dan is ook $\forall v\varphi$ een axioma, want: $\forall v\varphi$ is een *generalisering* van φ. Daarmee is $\langle\forall v\varphi\rangle$ een bewijs voor $\forall v\varphi$. We hebben dus: $\vdash \forall v\varphi$, en het basisgeval is afgewerkt.
- *Inductiestap*: De inductiehypothese is: als φ een bewijs heeft van lengte n (of kleiner), dan geldt $\vdash \forall v\varphi$. Merk op dat de inductiehypothese *niets* zegt over de lengte van het bewijs van $\forall v\varphi$. We moeten laten zien dat nu uit $\vdash \varphi$ en "φ heeft een afleiding van lengte $n+1$" ook volgt: $\vdash \forall v\varphi$. Dat gaat zo: of φ is een axioma, en we redeneren als in het basisgeval en klaar, of φ volgt via Modus Ponens uit ψ en $\psi \to \varphi$, waarbij op ψ en $\psi \to \varphi$ de inductiehypothese van toepassing is (ga na waarom). We hebben dus: $\vdash \forall v\psi$ en $\vdash \forall v(\psi \to \varphi)$. Hieruit leiden we $\forall v\varphi$ als volgt af:

1.	$\forall v(\psi \to \varphi) \to (\forall v\psi \to \forall v\varphi)$	[ax 6]
2.	$\forall v(\psi \to \varphi)$	[gegeven]
3.	$\forall v\psi \to \forall v\varphi$	[MP uit 1,2]
4.	$\forall v\psi$	[gegeven]
5.	$\forall v\varphi$	[MP uit 3,4].

  We hebben laten zien dat $\vdash \forall v\varphi$. Hiermee is de inductiestap compleet, en dus is het bewijs van het principe van Universele Generalisatie rond.

□

Het principe van Universele Generalisatie staat ons nu toe om uit

$$\vdash x = y \to y = x$$

(stelling 14.3) het volgende af te leiden:

$$\vdash \forall x \forall y (x = y \to y = x).$$

Met andere woorden: we hebben in de predikatenlogische calculus bewezen dat de identiteits-relatie *symmetrisch* is.

Een tweede voorbeeld van een metastelling voor de predikatenlogische calculus is de deductiestelling. Alle principes die we hebben gebruikt in het bewijs van de deductiestelling voor de propositielogica (vergelijk § 9.5) gaan ook op voor de predikatenlogische calculus, dus het eerder gegeven bewijs geldt ook hier.

Metastellingen als het principe van universele generalisatie en de deductiestelling geven ons in feite extra redeneerregels in handen. Op het feit dat de deductiestelling buitengewoon nuttig is als extra redeneerregel hebben we in § 9.5 al gewezen. Voor de algemene variant,

$$\text{als } \varphi_1, \ldots, \varphi_n, \varphi_{n+1} \vdash \psi \text{ dan } \varphi_1, \ldots, \varphi_n \vdash (\varphi_{n+1} \to \psi),$$

geldt dit in nog sterkere mate (zie opdracht 9.9 aan het eind van § 9.5). Stel dat we willen laten zien:

$$\vdash (\varphi \to (\psi \to \chi)) \to (\psi \to (\varphi \to \chi)),$$

dan is het volgens de deductiestelling genoeg om te laten zien:

$$\varphi \to (\psi \to \chi) \vdash \psi \to (\varphi \to \chi).$$

Om dat te laten zien is het, weer volgens de deductiestelling (maar nu in de algemene variant), genoeg om te bewijzen:

$$\varphi \to (\psi \to \chi), \psi \vdash \varphi \to \chi$$

en daarvoor is een bewijs van

$$\varphi \to (\psi \to \chi), \psi, \varphi \vdash \chi$$

weer genoeg. Dat bewijs gaat als volgt:

1.	$\varphi \to (\psi \to \chi)$	[gegeven]
2.	φ	[gegeven]
3.	$\psi \to \chi$	[MP uit 1,2]
4.	ψ	[gegeven]
5.	χ	[MP uit 3,4].

Je ziet het: heel wat gemakkelijker dan direct een bewijs leveren voor de formule waar we mee begonnen.

We zijn geïnteresseerd in de verzameling van alle formules die afleidbaar zijn uit een bepaalde aannamen-verzameling T in de predikatenlogische calculus. Wat dit betekent is dat we door het kiezen van geschikte uitgangspunten een specifiek *soort* van predikatenlogische modellen kunnen beschrijven. Om gemakkelijk over dit soort zaken te kunnen praten voeren we nieuw jargon in.

Definitie 14.5 *Een* **predikatenlogische theorie** T *is een verzameling van predikatenlogische formules.*

We noemen de formules uit T de *niet-logische axioma's* van de theorie. In feite leggen de niet-logische axioma's samen met de predikatenlogische axioma's (waarvoor we het recept hierboven hebben gegeven) en de afleidingsregel Modus Ponens de theorie vast.

Definitie 14.6 *De* **deductieve afsluiting** *van een theorie T, notatie $\overline{T}$, is de verzameling $\{\varphi \mid T \vdash \varphi\}$.*

De deductieve afsluiting van T is de verzameling van alle formules die met behulp van Modus Ponens kunnen worden afgeleid uit formules in T en logische axioma's.

Niet alle formuleverzamelingen T zijn even interessant. We zijn met name geïnteresseerd in theorieën T die *consistent* zijn (zie definitie 9.6 in § 9.6).

Opgave 14.16 Laat zien dat als T consistent is, dan $\overline{T}$ ook, en als $\overline{T}$ consistent is, dan T ook.

Een heel simpel voorbeeld van een predikatenlogische theorie is de theorie T die alleen de ene formule $\exists x\, x = x$ bevat, dus: $T = \{\exists x\, x = x\}$. Deze theorie is waar in alle predikaatlogische modellen met een niet-leeg domein: de formule $\exists x\, x = x$ is waar precies wanneer er minstens één ding bestaat. Dit sluit alleen het model met het lege domein uit. Merk op dat het model met het lege domein *niet* wordt uitgesloten door de definitie van 'model voor een predikatenlogische taal'.

Hier is een wat serieuzer voorbeeld van een predikatenlogische theorie, de zogenaamde theorie van de *partiële ordes*:

$$\begin{array}{l}\{\forall x Rxx,\\ \forall x\forall y((Rxy \wedge Ryx) \rightarrow x = y),\\ \forall x\forall y\forall z(Rxy \rightarrow (Ryz \rightarrow Rxz))\}.\end{array}$$

De drie formules die de theorie uitmaken drukken respectievelijk uit dat de relatie die door R wordt benoemd reflexief, antisymmetrisch en transitief is. Elk model voor deze theorie moet een partieel geordende verzameling zijn, dat wil zeggen een verzameling waarop een partiële orde is gedefinieerd (vergelijk § 3.1).

Een ander voorbeeld van een predikatenlogische theorie is de zogenaamde theorie van de *dichte onbegrensde lineaire ordes*. We laten de niet-logische axioma's die de theorie uitmaken een voor een de revue passeren:

1. $\forall x\forall y\forall z(Rxy \rightarrow (Ryz \rightarrow Rxz))$.

   Deze formule drukt uit dat de relatie die door R wordt benoemd *transitief* is.

2. $\forall x\forall y(Rxy \rightarrow \neg Ryx)$.

   Deze formule drukt uit dat de relatie die door R wordt benoemd *asymmetrisch* is.

3. $\forall x\forall y(Rxy \vee Ryx \vee x = y)$.

   Deze formule drukt uit dat de relatie die door R wordt benoemd de eigenschap van *samenhang* heeft (of: samenhangend is).

4. $\forall x\forall y(Rxy \rightarrow \exists z(Rxz \wedge Rzy))$.

   Deze formule drukt uit dat de relatie die door R wordt benoemd de eigenschap van *dichtheid* heeft (of: dicht is).

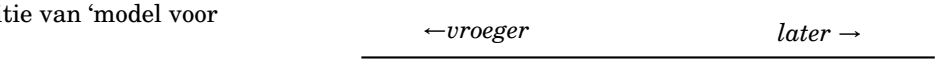
5. $\forall x\exists y Rxy$.

   Deze formule drukt uit dat de relatie die door R wordt benoemd *voortzetting naar rechts* kent.

6. $\forall x\exists y Ryx$.

   Deze formule drukt uit dat de relatie die door R wordt benoemd *voortzetting naar links* kent.

We kunnen nu gaan kijken naar *modellen* voor deze theorie. Een voorbeeld van zo'n model is de verzameling van alle *tijdstippen*, met R geïnterpreteerd als de relatie *eerder dan*. De formules van de theorie worden nu uitspraken over de tijd. Die uitspraken zijn als zodanig voor discussie vatbaar, maar dat is een andere kwestie. 1. wordt nu: als tijdstip 1 eerder valt dan tijdstip 2, en tijdstip 2 valt eerder dan tijdstip 3, dan valt tijdstip 1 eerder dan tijdstip 3. Net zo voor de andere formules. Het model voor de tijd dat hier beschreven wordt ziet er dus zo uit:



Vergelijk § 9.8, waar een dergelijke tijdsas figureert in de modellen voor de propositionele tijdslogica. Merk op dat in dit model de tijd zich onbegrensd uitstrekt naar verleden en toekomst, en dat er zich tussen elk tweetal tijdstippen, hoe dicht ze ook bij elkaar liggen, een derde bevindt. Op de filosofische implicaties van deze visie op tijd gaan we hier niet in.

Een ander model voor de theorie van de dichte onbegrensde lineaire ordes is het domein van alle positieve en negatieve breuken en het getal 0, met R geïnterpreteerd als de relatie *kleiner dan*. Dit model wordt vaak aangeduid als $\mathbb{Q}$. Ga zelf na dat formules 1. tot en met 6. waar zijn in $\mathbb{Q}$.

Andere voorbeelden van predikatenlogische theorieën zijn er te over. In de meeste gevallen hebben ze, in tegenstelling tot de bovengenoemde voorbeelden, niet een eindig maar een *oneindig* stel niet-logische axioma's. Het feit dat een theorie een oneindige verzameling niet-logische axioma's heeft is niet zo bezwaarlijk als op het eerste gezicht lijkt: we zijn met name geïnteresseerd in theorieën waarvan de axioma's *recursief* gedefinieerd kunnen worden. Als dat kan is de axioma-verzameling mogelijkerwijs wel oneindig, maar toch met eindige middelen weer te geven (net als de verzameling predikatenlogische axioma's). We zeggen dan dat zo'n oneindige theorie *oneindig recursief is geaxiomatiseerd*. Een beroemd voorbeeld is de theorie van de rekenkunde zoals geaxiomatiseerd in de axioma's van Peano (de zogenaamde Peano-rekenkunde). Deze theorie wordt vaak afgekort als PA.

Ook de verzamelingenleer kan in axiomatische vorm worden gepresenteerd. Dit axiomatiseren van de verzamelingenleer is ter hand genomen nadat Bertrand Russell een tegenspraak had afgeleid uit Georg Cantor's oorspronkelijke verzamelingenleer (de zogenaamde naïeve verzamelingenleer; vergelijk de titel van Hoofdstuk 2 van

dit dictaat). Het separatie-axioma, vermeld in § 4.8, ziet er nu bij voorbeeld als volgt uit:

$$\forall x \exists y \forall z (Rzy \leftrightarrow (Rzx \wedge \varphi(z))).$$

In deze formule is R een tweeplaatsige predikaatletter die wordt geïnterpreteerd als *is element van* (dat wil zeggen: de $\in$ relatie); $\varphi(z)$ is een formule waarin alleen de variabele z vrij voorkomt. In het axioma wordt $\varphi(z)$ gebruikt om de verzamelingen die voldoen aan φ te karakteriseren. Door een verstandige keuze van de overige axioma's kan nu worden bewerkstelligd dat de Russell paradox en verwante problemen niet meer optreden. Verschillende systemen zijn voorgesteld; het bekendste axiomastelsel is de zogenaamde Zermelo-Fraenkel verzamelingenleer (ingewijden korten dit af als ZF). Zie voor meer informatie het al eerder genoemde leerboek [Van Dalen e.a. 1975].

14.3 Correctheid, volledigheid, onbeslisbaarheid

Net als bij de propositielogica moeten we de brandende vraag beantwoorden naar de verhouding tussen de begrippen *afleidbaar* en *logisch gevolg*. Dus ten eerste: is de predikatenlogische calculus correct? Met andere woorden, geldt het volgende: als $\Gamma \vdash \varphi$, dan $\Gamma \models \varphi$ (waarbij Γ een willekeurige formuleverzameling is, en φ een willekeurige formule)? Om dit te laten zien moet eerst het feit uit de volgende opdracht worden aangetoond.

Opgave 14.17 Laat zien: als $b(v) = b'(v)$ voor elke vrije variable v in φ, dan $V_b(\varphi) = V_{b'}(\varphi)$.

Het bewijs van de correctheid van de predikatenlogische calculus heeft nogal wat voeten in de aarde.

Stelling 14.5 (Correctheid van de predikatenlogica)

Voor elke predikatenlogische formuleverzameling Γ en elke predikatenlogische formule φ: als $\Gamma \vdash \varphi$, dan $\Gamma \models \varphi$.

Bewijs: We plegen inductie naar de lengte van de afleiding van φ uit Γ, en maken daarbij gebruik van het resultaat uit opdracht 14.17. Één-staps-afleidingen zijn er in twee soorten:

- $\varphi \in \Gamma$. In dat geval geldt voor elk model $\mathcal{M} = \langle D, I \rangle$ met bijbehorende valuatie V en iedere bedeling b dat $V_b(\varphi) = 1$ als voor elke $\gamma \in \Gamma$ geldt $V_b(\gamma) = 1$. Dus is in dit geval $\Gamma \models \varphi$.
- φ is een predikatenlogisch axioma. Zo'n axioma is een generalisering van één van de schema's 1–6. De geldigheid hiervan kunnen we wederom bewijzen met inductie (naar het aantal universele kwantoren dat bij generalisering voorop geplaatst wordt):
  - De schema's 1–3 gelden nog steeds; weliswaar mogen de formules nu variabelen, kwantoren, predikaten, enzovoort bevatten, de geldigheid van deze schema's berust uitsluitend op de predikatenlogische valuatie van $\rightarrow$ en $\neg$ en die is in essentie hetzelfde als in de propositielogica.
  - De geldigheid van axiomaschema 4 is zeker niet triviaal. Stel voor willekeurige valuatie V en bedeling b met domein D dat $V_b(\forall v \varphi) = 1$, dan is voor ieder object $d \in D$: $V_{b(v|d)}(\varphi) = 1$. We moeten nu laten zien dat $V_b([t/v]\varphi) = 1$ mits t substitueerbaar is voor v in φ. Merk op dat als $d = W_b(t)$ dan

    $V_b([t/v]\varphi) = V_{b(v|d)}(\varphi)$ mits t substitueerbaar voor v in φ.

    Deelbewijs: opnieuw met inductie, ditmaal naar de opbouw van φ (het totale bewijs heeft dus de vorm van een 3-traps-inductie). Zowel de basisstap(pen) voor atomaire formules als de inductiestappen voor de connectieven zijn eenvoudig. De kwantorstap is echter wel complex:
    Neem voor φ de formule $\forall u \varphi$. We onderscheiden twee gevallen:
    - $v = u$**.** Dan $V_b([t/v]\forall u\varphi) = V_b(\forall u\varphi) = V_{b(u|d)}(\forall u\varphi) = V_{b(v|d)}(\forall u\varphi)$ (de voorlaatste stap gebruikt het resultaat uit opdracht 14.17);
    - $v \neq u$**.** Dan $V_b([t/v]\forall u\varphi) = V_b(\forall u[t/v]\varphi) = 1$ desda voor elke $e \in D$: $V_{b(u|e)}([t/v]\varphi) = 1$. De inductiehypothese (toegepast op bedeling $b(u|e)$) voor dit deelbewijs levert dat voor elke $e \in D$ $V_{b'}(\varphi) = 1$, waarbij $b' = b(u|e)(v|d)$ en substitueerbaar-zijn met zich meebrengt dat $t \neq u$, en derhalve $W_{b(u|e)}(t) = W_b(t) = d$. Omdat $v \neq u$ geldt echter ook dat $b' = b(v|d)(u|e)$ en daarom volgt de equivalentie met $V_{b(v|d)}(\forall u\varphi) = 1$, hetgeen te bewijzen was.
  - Axiomaschema 5 laat zich eenvoudig verifiëren: stel maar voor willekeurige D,V en b: $V_b(\varphi) = 1$ en v niet vrij in φ. Voor willekeurige $d \in D$ geldt dan volgens het resultaat uit opdracht 14.17 (immers b en $b(v|d)$ zijn dan identiek voor de vrije variabelen in φ): $V_{b(v|d)}(\varphi) = 1$, ergo $V_b(\forall v\varphi) = 1$.
  - De geldigheid van schema 6 bewijzen we indirect: stel dat 6 niet geldt. Dan is er een dus een model met valuatie V, domein D en bedeling b zodat (a) $V_b(\forall v(\varphi \rightarrow \psi) = 1$ en (b) $V_b(\forall v\varphi \rightarrow \forall v\psi) = 0$. Uit (b) volgt dat (c) $V_b(\forall v\varphi) = 1$ en (d) $V_b(\forall v\psi) = 0$. (d) impliceert dat er een $e \in D$ zou bestaan waarvoor $V_{b(v|e)}(\psi) = 0$. Uit (a) en (c) volgen echter respectievelijk dat $V_{b(v|e)}(\varphi \rightarrow \psi) = 1$ en $V_{b(v|e)}(\varphi) = 1$, en samen levert dit $V_{b(v|e)}(\psi) = 1$, wat in tegenspraak is met het gevolg van (d).
  - We bewijzen de semantische geldigheid van het principe van Universele Generalisatie: laat $\models \varphi$. Dan geldt voor elke valuatie V en bedeling b dat $V_b(\varphi) = 1$ en dus ook elke V en b: $V_{b(v|d)}(\varphi) = 1$ voor elk object d uit het domein. Dus geldt per definitie voor elke V en B: $V_b(\forall v\varphi) = 1$, kortom $\models \forall v\varphi$.

De geldigheid van Universele Generalisatie en de

schema's 1–6 heeft de geldigheid van alle predikatenlogische axioma's tot gevolg.

– De enige afleidingsregel van het formele systeem is Modus Ponens. Stel nu dat $\Gamma \models \varphi$ en $\Gamma \models \varphi \rightarrow \psi$. Kies een valuatie V en een bedeling b zodat $V_b(\gamma) = 1$ voor elke $\gamma \in \Gamma$. Uit de veronderstellingen volgt nu $V_b(\varphi) = 1$ en $V_b(\varphi \rightarrow \psi) = 1$, en dus $V_b(\psi) = 1$. Maar dan $\Gamma \models \psi$. □

Dat was het bewijs van de predikatenlogische correctheid. Ten tweede: is de predikatenlogische calculus volledig, in de zin dat het volgende geldt: als $\Gamma \models \varphi$, dan $\Gamma \vdash \varphi$ (weer met Γ een willekeurige formuleverzameling en φ een willekeurige formule) ? De logicus Kurt Gödel bewees in 1930 dat dit inderdaad het geval is. Deze stelling heet: de volledigheidsstelling van Gödel.

Wellicht vraagt de lezer zich af hoe Gödel in 1930 een eigenschap van een systeem kon bewijzen die Tarski eerst in 1933 voorstelde. De verklaring van dit schijnbare anachronisme is dat de ideeën over modeltheoretische semantiek al geruime tijd circuleerden in werk van bij voorbeeld Hilbert, Bernays en Ackermann. Tarski heeft aan die gedachten een precieze en algemene vorm gegeven, maar voor Gödel waren de eerdere voorstellen al voldoende. Zijn bewijs maakte geen gebruik van de methode van Henkin, maar wel van de in § 11 genoemde stelling van Löwenheim en Skolem. Wij laten een bewijs van de volledigheid van de predikatenlogica hier overigens achterwege.

Een alternatieve formulering voor de volledigheidsstelling van Gödel is de volgende: elke (syntactisch) consistente verzameling formules is vervulbaar (in enig model). We laten zien dat dit volgt uit de gebruikelijke formulering van de volledigheidsstelling. Consistentie van een formuleverzameling Γ komt—zoals we in stelling 9.7 bewezen hebben—neer op:

$$\text{Er is een formule } \varphi \text{ zo dat } \Gamma \nvdash \varphi.$$

Met behulp van de volledigheidsstelling volgt hieruit, door toepassen van contrapositie:

$$\Gamma \not\models \varphi.$$

Wat wil dit zeggen? Niets anders dan: er is minstens één model $\mathscr{M}$ en een bedeling b waarin elke formule uit Γ vervulbaar is, maar φ niet. Met andere woorden: Γ is vervulbaar.

Deze herformulering van Gödels volledigheidsstelling maakt duidelijk hoe de syntactische definitie van *consistentie* uit 14.2 kan leiden tot de semantische karakterisering van *inconsistente verzameling formules* als: verzameling formules die niet vervulbaar is.

Een ander punt is de *beslisbaarheid* van de predikatenlogica. Een theorie T is *beslisbaar* wanneer er een mechanische procedure bestaat om, bij een gegeven formule φ van de taal, uit te maken of $T \vdash \varphi$ dan wel $T \nvdash \varphi$. De vraag naar de beslisbaarheid van de predikatenlogica is een bijzonder geval: het geval waar T leeg is. Dus: is er een mechanische procedure om bij een gegeven formule φ van de taal uit te maken of φ een stelling is? In 12 hebben we gezien dat de methode van de semantische tableaus voor de predikatenlogica geen beslissingsprocedure is. Dat zat hem in het feit dat de tableau-regels voor de kwantoren herhaald moeten worden toegepast, zodat er geen garantie meer is dat een tableau in eindig veel stappen kan worden voltooid. We kunnen nu direct het volgende inzien.

Stelling 14.6 *De verzameling van alle* **kwantorvrije stellingen** *van de predikatenlogica is beslisbaar.*

Bewijs: Wanneer φ een kwantorvrije predikatenlogische formule is, dan komen er aan het tableau alleen propositielogische tableauregels te pas, regels die variabelen en constanten vervangen door 'standaardnamen' uit het rijtje $d_1, d_2, \ldots$ plus eventueel de regels voor identiteit. De regel voor =-links moet herhaald worden toegepast, maar het aantal malen is eindig, want het aantal voorkomens van standaard-namen uit $d_1, d_2, \ldots$ in het tableau is beperkt tot namen die zijn ingevoerd voor variabelen en constanten in de oorspronkelijke formule φ, en dat zijn er eindig veel. Hieruit zien we dat het tableau voor φ in eindig veel stappen is voltooid. Vanwege het feit dat de predikatenlogica volledig is mogen we deze beslissingsprocedure voor $\models$ beschouwen als een beslissingsprocedure voor $\vdash$. □

Aantonen dat de predikatenlogica als geheel niet beslisbaar is heeft veel meer voeten in de aarde: uit het feit dat de tableau-methode geen beslissingsmethode is volgt uiteraard nog niet dat er niet een andere testmethode zou kunnen zijn die *wel* een beslissingsmethode is.

Hoewel de predikatenlogica als geheel niet beslisbaar is het heel wel mogelijk om in de predikatenlogica beslisbare *theorieën* te formuleren: de extra niet-logische axioma's van de theorie perken de klasse van modellen dusdanig in dat de notie $\models$ (en dus $\vdash$) wat hanteerbaarder wordt, zou je kunnen zeggen. Een voorbeeld van een beslisbare predikatenlogische theorie is de theorie van de dichte onbegrensde lineaire ordes (vergelijk § 14.2). Bij de theorie van de Peano-rekenkunde (PA) en de Zermelo-Fraenkel axiomatisering van de verzamelingenleer (ZF) ligt het anders. Deze theorieën zijn *niet* beslisbaar. Een C#-programma dat uitmaakt of een willekeurige bewering φ afleidbaar is uit PA of uit ZF valt niet te schrijven.

De welgevormde formules van een predikatenlogische taal kunnen worden opgesomd. Het zijn immers eindige rijtjes tekens in een eindig alfabet, en zodra er een volgorde voor dat alfabet is vastgelegd kunnen alle formules van de taal in de in hoofdstuk 3 besproken telefoonboek-volgorde achter elkaar worden gezet. Dit levert een gigantisch maar aftelbaar telefoonboek op. Neem aan dat de formules genummerd zijn als 0, 1, 2, 3 enzovoort: met iedere formule correspondeert dus een natuurlijk getal. Het volgende is nu vrij eenvoudig in te zien.

Stelling 14.7 *De verzameling van de natuurlijke getallen die corresponderen met de stellingen van de predikatenlogica is opsombaar.*

Bewijs: We mogen de bewering uit de stelling wel parafraseren als: "Er bestaat een computerprogramma dat de verzameling *stellingen* van de predikatenlogica opsomt". Het gezochte programma moet dus elke formule die een stelling is, of het nummer van die formule in het formules-telefoonboek, vroeg of laat afdrukken.

Dat er zo'n programma bestaat blijkt als volgt. Een stelling van de predikatenlogica is de laatste formule in een *bewijs*, en een bewijs is een eindig rijtje formules dat aan zekere eisen voldoet. De verzameling van alle eindige rijtjes formules is weer *aftelbaar* (gebruik het telefoonboek dat alle formules opsomt om een nieuw telefoonboek te maken dat alle eindige *rijtjes* van formules opsomt). Ons computerprogramma werkt nu als volgt (commentaar staat tussen { }):

BEGIN
maak n gelijk aan 0;
(1) controleer of het formulerijtje dat positie n inneemt een bewijs is;
{ deze controle kan mechanisch worden verricht: er hoeft alleen maar te worden gecontroleerd of de formules in het rijtje axioma's zijn, dan wel resultaten van het toepassen van MP op twee voorafgaande formules }
ALS formulerijtje met nummer n een bewijs is
DAN druk de laatste formule in het rijtje af;
{ want dat is een stelling }
maak n gelijk aan $n+1$;
GA NAAR (1);
EINDE.

Het is duidelijk dat alle stellingen zo vroeg of laat te voorschijn komen. □

Merk op dat het programma dat in de stelling beschreven wordt nooit stopt: het blijft tot in lengte van dagen formules uitbraken.

Je ziet het: de opsombaarheid van de verzameling stellingen van de predikatenlogica hangt samen met de mogelijkheid om *mechanisch te controleren* of een gegeven rijtje formules een bewijs is voor een 'kandidaat-stelling'. Als we vragen of de verzameling stellingen van de predikatenlogica beslisbaar is vragen we (veel) meer: we vragen dan of er ook een mechanische methode is om bewijzen te *vinden*. Alonzo Church heeft in 1936 bewezen dat het antwoord op deze vraag *nee* is.

Stelling 14.8 (Stelling van Church) *Er bestaat geen recursieve procedure die de verzameling stellingen van de predikatenlogica herkent.*

Bewijs: Zie [Van Dalen 1983] of [Enderton 1972]. □

Met behulp van de these van Church volgt uit deze stelling: de predikatenlogica is *niet beslisbaar*. De stelling van Church leidt meteen tot de volgende stelling.

Stelling 14.9 *De verzameling van niet-stellingen van de predikatenlogica is* **niet** *opsombaar.*

Bewijs: Stel dat de verzameling van niet-stellingen van de predikatenlogica wel opsombaar zou zijn. Dan zouden we een computerprogramma hebben dat de verzameling formules opsomt die *geen* stellingen zijn van de predikatenlogica. Noem dit het niet-stellingen-programma. We hadden ook al een programma dat de stellingen van de predikatenlogica opsomde (zie stelling 14.7). Noem dit programma het stellingen-programma. Deze twee programma's kunnen nu worden gecombineerd tot een beslissingsprogramma dat voor willekeurige formules nagaat of het stellingen zijn of niet. In het beslissingsprogramma hieronder is φ een variabele die de formule bevat die we onderzoeken. We maken gebruik van de notatie := voor het toekennen van een waarde aan een variabele; $n := 0$ staat dus voor "maak de waarde van n gelijk aan 0". *WelEenStelling* en *GeenStelling* zijn twee Boolese variabelen, dat wil zeggen: variabelen die een waarheidswaarde bevatten.

BEGIN
n := 0;
WelEenStelling := onwaar;
GeenStelling := onwaar;
HERHAAL
ALS φ gelijk is aan formule n in de uitvoer van het stellingen-programma DAN
WelEenStelling := waar
ANDERS ALS φ gelijk is aan formule n in de uitvoer van het niet-stellingen-programma DAN
GeenStelling := waar;
n := n + 1
TOTDAT WelEenStelling of GeenStelling;
ALS WelEenStelling DAN schrijf (φ, ' is een stelling.')
ANDERS schrijf (φ, ' is geen stelling.')
EINDE.

Het bestaan van zo'n beslissingsprogramma is in strijd met de stelling van Church, dus de aanname dat de verzameling niet-stellingen van de predikatenlogica kan worden opgesomd moet worden verworpen. □

14.4 Volledige en onvolledige theorieën

Om nog iets naders te kunnen zeggen over de eisen waaraan een predikatenlogische theorie moet voldoen om beslisbaar te zijn voeren we nu een nieuw begrip in.

Definitie 14.7 *Een theorie T heet* **volledig** *wanneer voor elke zin φ van de taal geldt: $T \vdash \varphi$ of $T \vdash \neg\varphi$.*

Merk op dat deze definitie betrekking heeft op de *zinnen* (dat wil zeggen: de gesloten formules) van een theorie. Het is niet redelijk om van een theorie te verwachten dat ook voor open formules (zoals Rxy) geldt dat ofwel de formule zelf ofwel zijn negatie afleidbaar is. We hoeven

niet bang te zijn dat er door deze beperking interessante formules uit de boot vallen, want (zoals je gemakkelijk kunt nagaan): als een of andere open formule φ afleidbaar is uit een theorie T, dan is zijn *universele afsluiting* (dat wil zeggen: het resultaat van universeel kwantificeren over alle variabelen die vrij in φ voorkomen) ook afleidbaar uit T.

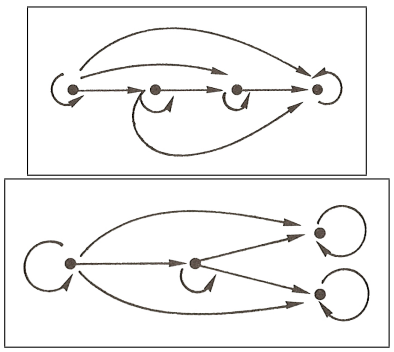
Was de eis van consistentie een soort minimum-eis die aan theorieën kan worden gesteld, *volledigheid* is een maximum-eis, en er is slechts een select gezelschap van theorieën dat eraan voldoet.

Opgave 14.18 Bewijs dat voor wat betreft gesloten formules:

$$T \text{ is consistent en volledig} \Leftrightarrow \overline{T} \text{ is maximaal consistent.}$$

Let op: het begrip *volledigheid* dat hier wordt geïntroduceerd is een *ander* begrip dan de notie die we in de paragrafen 9.6 en 14.3 hebben gebruikt. Het is ook verschillend van het begrip *functionele volledigheid* uit § 8.5. De predikatenlogica is *volledig* in de zin dat predikatenlogische geldigheid predikatenlogische bewijsbaarheid impliceert, maar de predikatenlogische theorie $\emptyset$ is niet volledig in de zin van de zojuist gegeven definitie van volledigheid voor theorieën.

De theorie van de partiële ordes is ook niet volledig. Hier zijn twee voorbeelden van structuren die aan de niet-logische axioma's van de theorie voldoen; in de ene is de zin $\forall x \forall y(Rxy \vee Ryx)$ waar, in de andere niet. Beide structuren zijn partiële ordes.



Kennelijk zit noch $\forall x \forall y(Rxy \vee Ryx)$ noch $\neg\forall x \forall y(Rxy \vee Ryx)$ in de deductieve afsluiting van de theorie van de partiële ordes. Probeer zelf na te gaan waar deze conclusie op berust.

We stellen hier zonder bewijs dat de theorie van de dichte onbegrensde lineaire ordes volledig is. Voor het bewijs is een modeltheoretische exercitie nodig die buiten het bestek van dit dictaat valt. Hieruit, in combinatie met het feit dat de theorie in kwestie een eindige verzameling axioma's heeft, volgt de beslisbaarheid van de theorie, met behulp van de volgende stelling.

Stelling 14.10 *Als theorie T eindig geaxiomatiseerd is en volledig, dan is T beslisbaar.*

Bewijs: Omdat T eindig geaxiomatiseerd is mogen we aannemen dat de verzameling

$$\{\varphi \mid \varphi \text{ is een gesloten formule en } T \vdash \varphi\}$$

opsombaar is. Er is dus een programma dat de gesloten formules in de deductieve afsluiting van T opsomt. Uit de volledigheid van T volgt nu dat het volgende programma een beslissingsprogramma is voor de gesloten formules die uit T afleidbaar zijn. Neem weer aan dat φ een variabele is die een ingelezen formule bevat; *WelAfleidbaar* en *NietAfleidbaar* zijn Boolese variabelen):

```
BEGIN
    n := 0;
    WelAfleidbaar := onwaar;
    NietAfleidbaar := onwaar;
    HERHAAL
        ALS φ gelijk is aan formule n in de uitvoer
        van het opsom-programma DAN
            WelAfleidbaar := waar;
        ANDERS BEGIN
            {in de nu volgende opdracht slaat
            'concatenatie' op het aan elkaar
            plakken van schrijftekens: }
            ALS concatenatie('¬', φ) gelijk is
            aan formule n in de uitvoer
            van het opsom-programma DAN
                NietAfleidbaar := waar;
        EINDE;
        n := n + 1
    TOTDAT WelAfleidbaar of NietAfleidbaar;
    ALS WelAfleidbaar DAN
        schrijf ( φ, ' is afleidbaar.')
    ANDERS schrijf (φ, ' is niet afleidbaar.')
EINDE.
```

Hiermee is het bewijs van de stelling geleverd. □

Deze stelling laat zien dat volledige theorieën, mits ze overzichtelijk zijn geaxiomatiseerd (in feite hoeft de verzameling axioma's niet eindig te zijn, als ze maar beslisbaar is), 'geknipt zijn voor de computer'. Echter, zulke theorieën zijn schaars. Voor logici en wiskundigen is dat trouwens maar gelukkig ook, want anders zou er, als we de computers eenmaal geprogrammeerd hebben voor de beslisprocedures van de wiskundige theorieën die ons interesseren, voor hen weinig meer te doen zijn.

Zoals Kurt Gödel in 1931 bewees: PA en ZF zijn *niet* volledig (mits ze consistent zijn). Dit zijn de beroemde *onvolledigheidsresultaten* van Gödel. De precieze gedachtegang vind je weer in [Enderton 1972]. In [Nagel & Newman 1975] en [Hofstadter 1979] staan meer populaire versies. We geven een korte samenvatting.

PA, de Peano-axiomatisering van de rekenkunde, is een verzameling axioma's die bedoeld is om de structuur bestaande uit de verzameling $\mathbb{N}$, met daarop de rekenkundige operaties, te beschrijven. Deze structuur duiden we voor het gemak aan met $\mathbb{N}$. Om aan te tonen dat PA onvolledig is, aangenomen dat deze theorie

consistent is, construeert Gödel een zin ψ zo dat $\mathrm{PA} \nvdash \psi$, en zo dat ψ *waar* is voor de natuurlijke getallen (dat wil zeggen: $\mathbb{N} \models \psi$). Waarom impliceert het bestaan van deze ψ dat PA onvolledig is? Dat zit zo. Uit de volledigheid van de predikatenlogica volgt: als $\mathrm{PA} \vdash \neg\psi$, dan $\mathrm{PA} \models \neg\psi$. Maar vanwege het feit dat $\mathbb{N} \models \mathrm{PA}$ (alle Peano-axioma's zijn waar op de natuurlijke getallen) hebben we: $\mathbb{N} \models \neg\psi$, en dit is in strijd met het gegeven dat ψ waar was voor de natuurlijke getallen. Dus kan $\neg\psi$ niet afleidbaar zijn uit PA. Een voor de hand liggende opmerking is nu: blijkbaar was PA te 'zuinig' als axiomatisering van de rekenkunde; laten we gewoon ψ aan de verzameling niet-logische axioma's toevoegen, en klaar. Dit lukt echter niet. Gödels constructievoorschrift is zo universeel dat voor $\mathrm{PA} \cup \{\psi\}$, de nieuwe theorie, weer een nieuwe voortvluchtige formule ψ' te construeren valt zodanig dat $\mathrm{PA} \cup \{\psi\} \nvdash \psi'$ en $\mathbb{N} \models \psi'$, enzovoorts.

Het basisidee achter de constructie van Gödels zin ψ stamt uit de filosofie van de Oudheid. Gödels zin houdt verband met de zogenaamde 'Leugenaarparadox' van de stoïcijnen (±300 voor Christus). Epimenides de Cretenzer zegt: "Alle Cretenzers liegen (altijd)". Deze bewering kan niet waar zijn. Immers, als Epimenides de waarheid spreekt, dan kunnen we uit de inhoud van zijn uitspraak afleiden dat Epimenides—zelf een Cretenzer—altijd liegt, dus dan is zijn uitspraak niet waar, en hebben we een tegenspraak. De uitspraak van Epimenides weerlegt dus zichzelf. Dit is een voorbeeld van wat we een 'halve' paradox zouden kunnen noemen, in tegenstelling tot een 'volkomen' paradox als de Russell-paradox die we in § 14.2 zijn tegengekomen. Bij een 'volkomen' paradox leidt zowel de aanname dat zekere bewering waar is als de aanname dat ze onwaar is tot een tegenspraak. Gödel had voor zijn onvolledigheids-redenering een halve paradox nodig omdat hij één vluchtweg wilde afsluiten: de halve paradox dient om de mogelijkheid uit te sluiten dat de formule ψ waar het verhaal om draait afleidbaar is in PA.

Opgave 14.19 Laat zien dat "Dit dictaat bevat fouten" een voorbeeld is van een halve paradox.

Het is overigens niet moeilijk de leugenaar-paradox zo te herformuleren dat het een volkomen paradox wordt. "Ik lieg nu", en "Deze zin is onwaar" zijn voorbeelden van volkomen paradoxen.

Opgave 14.20 Laat zien dat de bewering "Deze zin is onwaar" een volkomen paradox is.

Liefhebbers van paradoxen moeten
[Hughes & Becht 1978] lezen; fraaie puzzelvariaties op de leugenaarparadox zijn te vinden in [Smullyan 1978] en [Smullyan 1982]. Dit laatste boek biedt een smeuïge en informele inleiding in de achtergronden van Gödel's onvolledigheids-resultaten.

Gödels bewering ψ maakt gebruik van het feit dat de theorie van de rekenkunde rijk genoeg is om—door middel van een geschikte *codering*—met behulp van rekenkundige zinnen te kunnen praten over de *bewijsbaarheid* van andere rekenkundige zinnen. De leugenaarparadox krijgt nu de volgende vorm: ψ drukt uit dat *hijzelf* niet in PA afleidbaar is. Met andere woorden:

$$\psi = \text{"}\mathrm{PA} \nvdash \psi\text{"}.$$

Als $\mathrm{PA} \vdash \psi$, dan is ψ waar (dat wil zeggen $\mathbb{N} \models \psi$), want de stellingen van PA zijn ware beweringen over de natuurlijke getallen, en dus weten we, op grond van wat ψ uitdrukt, dat $\mathrm{PA} \nvdash \psi$, en we hebben een tegenspraak. Dus moet gelden:

$$\mathrm{PA} \nvdash \psi.$$

Maar dit is nu juist wat ψ zelf uitdrukte. Vanwege de volledigheid van de predikatenlogica volgt hieruit: ψ is waar. Als ψ waar is, dan kan $\neg\psi$ niet waar zijn, en wat niet waar is, is niet bewijsbaar (anders zouden de PA-axioma's niet *correct* zijn). Dus hebben we ook:

$$\mathrm{PA} \nvdash \neg\psi.$$

We besluiten met een opdracht die een variant is op een raadsel uit [Smullyan 1982]:

Opgave 14.21 Een *Gödel-schrijfmachine* is een schrijfmachine die door middel van het afdrukken van symbolen beweringen doet over zijn eigen afdruk-mogelijkheden. We nemen aan dat de machine correcte uitspraken doet over wat hij kan afdrukken. Met andere woorden: wanneer de machine zegt dat hij een rijtje symbolen wel of niet kan afdrukken, dan is dat ook zo. De machine gebruikt alleen de tekens `A`, `N` en `V`. Om te kunnen omschrijven wat de betekenis is hebben we een metavariabele X nodig voor rijtjes symbolen uit {`A`,`N`,`V`}. Als X een rijtje is, dan noemen we XX de verdubbeling van X. De betekenis van de rijtjes die de machine afdrukt wordt gegeven door de volgende clausules:

- `AX` is waar desda het rijtje X afdrukbaar is.
- `NAX` is waar desda het rijtje X niet afdrukbaar is.
- `VAX` is waar desda de verdubbeling van X afdrukbaar is.
- `VNAX` is waar desda de verdubbeling van X niet afdrukbaar is.

Wat is nu de simpelste ware bewering, uitgedrukt als een rijtje van tekens uit {`A`,`N`,`V`}, die de machine *niet* kan afdrukken?

Hoofdstuk 15

Uitbreidingen van de predikatenlogica

15.1 Meersoortige predikatenlogica

In de semantiek van de predikatenlogica zoals we die in § 11 gepresenteerd hebben was sprake van een domein van individuen. Alle variabelen werden toebedeeld aan objecten in dit ene domein. In de standaard-predikatenlogica zijn—zo zou je kunnen zeggen—alle objecten waarover wordt gekwantificeerd van hetzelfde soort. Wanneer je onderscheid wilt maken moet je dat doen met behulp van een geschikt predikaat. Neem de volgende twee zinnen:

Sommige mensen komen ongelegen.

Sommige tijdstippen komen ongelegen.

Neem als vertaalsleutel: Mx: x is een mens; Tx: x is een tijdstip; Oxy: x komt ongelegen op moment y; het discussiedomein is: mensen en tijdstippen. De predikatenlogische vertalingen worden nu, respectievelijk:

$\exists x(Mx \wedge \exists y Oxy)$.

$\exists y(Ty \wedge \exists x Oxy)$.

Door deze manier van onderscheidingen maken met behulp van predikaten kunnen de vertalingen behoorlijk ingewikkeld worden.

Iedereen komt soms ongelegen, maar sommigen komen altijd ongelegen. (1)

De vertaling wordt:

$$\forall x(Mx \rightarrow \exists y(Ty \wedge Oxy)) \wedge \exists z(Mz \wedge \forall u(Tu \rightarrow Ozu)).$$

Soms helpen vierkante haakjes:

$$\forall x[Mx \rightarrow \exists y(Ty \wedge Oxy)] \wedge \exists z[Mz \wedge \forall u(Tu \rightarrow Ozu)].$$

Accolades (“{”, “}”) zijn minder geschikt als alternatieve haakjes, die gebruiken we (per conventie) liever voor verzamelingen. Brackets (“⟨”, “⟩”) zou weer wel kunnen, mits er geen geordende paren in de formule voorkomen.

De ingewikkeldheid van de vertalingen is op zichzelf niet zo erg—een kwestie van wennen, nietwaar—maar er is nog een andere moeilijkheid: je zou kunnen zeggen dat de formule Oxy (voor: ‘x komt ongelegen op y’) alleen zinnig kan worden geïnterpreteerd wanneer het eerste argument van het predikaat O een mens is en het tweede een tijdstip. Immers: ‘Vastenavond komt ongelegen op Jan’ is niet gewoon onwaar, het is gewoon onzin. Het is een beetje vervelend dat dit soort onzin *wel* predikatenlogisch kan worden uitgedrukt, gegeven deze vertaalsleutel. ‘Jan komt ongelegen op vastenavond’ kan daarentegen zowel waar als onwaar zijn. (Overigens moet *vastenavond* eigenlijk eerder geïnterpreteerd worden als een *tijdsspanne* dan als een tijdstip. We zullen daar echter nu niet moeilijk over doen.)

Remedie: stap over van de gewone predikatenlogica naar *meersoortige* predikatenlogica. In plaats van een individuendomein hebben we nu meerdere individuendomeinen, een voor elke ‘soort’ van individuen. Bij kwantificatie moet nu worden aangegeven over welke soort van individuen de kwantoren lopen. Dat kan op twee manieren (en het maakt niets uit welke manier we kiezen): we kunnen bij iedere kwantor met behulp van een *index* aangeven over welke *soort* hij loopt, of we kunnen elke soort zijn eigen variabelen-verzameling geven. We geven een schets van beide werkwijzen voor een predikatenlogische taal met twee soorten: de soort *mens* en de soort *tijdstip*.

Eerste notatievariant: de kwantoren $\forall, \exists$ worden vervangen door $\forall_m$, $\exists_m$, $\forall_t$, $\exists_t$. De verzameling variabelen blijft: $x, y, z, u, v, \ldots$ De vertaling van (1) wordt nu:

$\forall_m x \exists_t y Oxy \wedge \exists_m z \forall_t u Ozu$.

Nog een paar vertalingen:

Sommige mensen komen soms ongelegen $\rightsquigarrow \exists_m x \exists_t y Oxy$.

Altijd is Kortjakje ziek $\rightsquigarrow \forall_t x Zkx$.

Voor dit laatste voorbeeld is er nog een preciezer vertaling mogelijk, voor het geval je ook nog expliciet zou willen uitdrukken dat k de naam is van een mens. Dat kan met een trucje:

$\exists_m x \forall_t y(k = x \wedge Zxy)$.

Opgave 15.1 Vertaal op deze manier, na een discussiedomein met meerdere soorten van objecten en een vertaalsleutel te hebben gegeven:

1. Soms is Kortjakje ziek, maar niet altijd.
2. Elke zondag gaat Kortjakje naar een kerk.
3. Iedereen is soms ziek.

Tweede notatievariant: de kwantoren $\forall$ en $\exists$ blijven gehandhaafd, maar we voeren twee verschillende verzamelingen van variabelen in:

– $x, x', x'', \ldots$ voor mensen;

– $t, t', t'', \ldots$ voor tijdstippen.

De vertalingen worden nu, respectievelijk:

$$\forall x \exists t O x t \wedge \exists x' \forall t' O x' t' \ ; \ \exists x \exists t O x t \ ; \ \forall t Z k t.$$

Het Kortjakje-voorbeeld maakt duidelijk dat we ook voor de constanten van de taal zouden moeten aangeven in welke soort ze geïnterpreteerd moeten worden. Voor de predikaatletters geldt iets analoogs: voor elke predikaatletter moet voor elke argumentplaats worden vastgelegd over welke soort hij loopt. Voorbeeld: Oxy voor '(mens) x komt ongelegen op (tijdstip) y' zou eigenlijk moeten worden genoteerd als $O_{\langle m,t\rangle} xy$, om aan te geven dat de eerste argumentplaats in de soort *mens* moet worden geïnterpreteerd, de tweede in de soort *tijdstip*. Jargon: het paar $\langle m, t\rangle$ wordt wel een *soort-type* genoemd.

Opgave 15.2 Vertaal de zinnen uit opdracht 15.1 op de manier van deze tweede notatiewijze.

We zullen de semantiek voor meersoortige predikatenlogica hier niet helemaal uitspellen. Globaal komt het er op neer dat het domein D wordt opgesplitst in subdomeinen voor de verschillende soorten. Dus voor het mensen-tijdstippen voorbeeld zou de structuur er zo uitzien $\langle D_m, D_t, I\rangle$. D_m zijn de mensen, D_t de tijdstippen. We mogen wel aannemen dat geen enkel object zowel een mens als een tijdstip is: de doorsnede van D_m en D_t is leeg. Dit laatste kan overigens van geval tot geval verschillen; in het algemeen hoeft het niet zo te zijn dat de subdomeinen voor de verschillende soorten disjunct zijn.

In systemen voor het 'vertalen' van natuurlijke taal zinnen in logische formules die momenteel worden ontwikkeld in het kader van Kunstmatige Intelligentie onderzoek speelt het gebruik van soort-type restricties een belangrijke rol bij het uitsluiten van niet-plausibele lezingen van zinnen. Zoals reeds opgemerkt zijn formules waarin gezondigd wordt tegen bepaalde soort-type restricties, bij voorbeeld de formule die de vertaling is van "Vastenavond komt ongelegen op Jan", eerder onzinnig dan onwaar. Dit feit kan worden verdisconteerd door de valuatiefunctie *partieel* te maken (zie § 3.2). Bij voorbeeld voor een atomaire formule als $O_{\langle m,t\rangle} ab$ geldt dan:

$$\begin{array}{lcl} V(O_{\langle m,t\rangle} ab) = 1 & \iff & \langle I(a), I(b)\rangle \in I(O_{\langle m,t\rangle}) \\ V(O_{\langle m,t\rangle} ab) = 0 & \iff & \langle I(a), I(b)\rangle \notin I(O_{\langle m,t\rangle}) \\ & & \ \& \ I(a) \in D_m \ \& \ I(b) \in D_t \\ V(O_{\langle m,t\rangle} ab) = \# & \iff & I(a) \notin D_m \ \text{ of } \ I(b) \notin D_t. \end{array}$$

De valuatie van de connectieven gaat dan volgens de zwakke waarheidstafels uit § 9.8. In het vervolg zullen we omwille van de eenvoud de valuatiefunctie gewoon *totaal* houden.

De interpretatiefunctie I moet aan de eisen voldoen die we hierboven hebben aangestipt: constanten die namen zijn voor mensen moeten interpretaties in D_m hebben, interpretaties voor namen van tijdstippen moeten juist in D_t zitten. Predikaatletters moeten worden geïnterpreteerd in overeenstemming met hoe de argumentplaatsen geïndiceerd zijn voor de soorten (bij voorbeeld $O_{\langle m,t\rangle}$ dient geïnterpreteerd te worden als een relatie tussen D_m en D_t). De identiteits-relatie = dient op elke soort als identiteit te worden geïnterpreteerd; in ons voorbeeld: we moeten onderscheiden tussen $=_{\langle m,m\rangle}$, te interpreteren als de identiteitsrelatie op D_m, en $=_{\langle t,t\rangle}$, te interpreteren als de identiteitsrelatie op D_t.

Is er nu, met de overgang van enkelsoortige naar meersoortige logica, essentieel iets veranderd? Met andere woorden: worden bepaalde gevolgtrekkingen geldig die dat vroeger niet waren of andersom? Het antwoord is: 'Nee'. Formules van een meersoortige predikatenlogische taal kunnen altijd worden *terugvertaald* naar formules van een enkelsoortige predikatenlogische taal met voor elke soort een eenplaatsige predikaatletter. In het voorbeeld: voer gewoon M weer in voor 'is een mens' en T voor 'is een tijdstip'. De terugvertaal-instructies luiden dan:

$$\forall_m x \ldots \rightsquigarrow \forall x(Mx \rightarrow \ldots)$$

$$\forall_t x \ldots \rightsquigarrow \forall x(Tx \rightarrow \ldots)$$

$$\exists_m x \ldots \rightsquigarrow \exists x(Mx \wedge \ldots)$$

$$\exists_t x \ldots \rightsquigarrow \exists x(Tx \wedge \ldots).$$

Opgave 15.3 Vertaal alle voorbeeld-formules uit de meersoortige logica die je in deze paragraaf bent tegengekomen terug naar de gewone, enkelsoortige predikatenlogica, met behulp van bovenstaande terugvertaal-instructies.

De formules die we zo krijgen kunnen worden geïnterpreteerd in de modellen die ontstaan door de subdomeinen van een meersoortig model samen te voegen en de soort-predikaatletters te interpreteren als de vroegere subdomeinen. Weer voor het voorbeeld: laat $\langle D_m, D_t, I\rangle$ een meersoortig model zijn. Dan is $\langle D_m \cup D_t, I^*\rangle$ het corresponderende enkelsoortige model. Hierbij duidt I^* de interpretatiefunctie aan die is als I, maar uitgebreid met een interpretatie voor M en T, namelijk $I^*(M) = D_m$ en $I^*(T) = D_t$. Als we de terugvertaling van een meersoortige formule φ noteren als φ^*, en het enkelsoortige model dat correspondeert met het meersoortige model $\mathcal{M}$ als $\mathcal{M}^*$, dan hebben we:

$$\mathcal{M} \models \varphi \iff \mathcal{M}^* \models \varphi^*.$$

Hieruit blijkt dat meersoortige predikatenlogica niet meer is dan een *variant* van de gewone predikatenlogica. Het gebruik ervan kan weleens *handiger* zijn dan dat van gewone predikatenlogica, maar dat is dan ook alles.

Een andere manier om een effect te krijgen als van meersoortige logica, maar met een nog kleinere afwijking van de gewone predikatenlogica, is door het invoeren per definitie van *beperkte kwantificatie*. Men neme een predikatenlogische taal en men spreke af: voor elke eenplaatsige predikaatletter A en elke formule φ van de

taal is $\forall x \in A : \varphi$ een afkorting voor $\forall x(Ax \to \varphi)$, en $\exists x \in A : \varphi$ een afkorting voor $\exists x(Ax \wedge \varphi)$. Omdat we alleen maar nieuwe afkortingen hebben ingevoerd verandert er niets aan de semantiek. Een paar vertalingen ter illustratie.

Alle mensen zijn sterfelijk $\rightsquigarrow \quad \forall x \in M : Sx.$

Sommige mensen komen altijd ongelegen
$\rightsquigarrow \quad \exists x \in M \forall y \in T : Oxy.$

Je begrijpt hopelijk dat de keuze 'beperkte kwantificatie of niet' een zuiver *notationele* keuze is. Beperkte kwantificatie is linguïstisch gezien wel handig omdat de bijdrage van nominale constituenten als geheel in de vertalingen nu beter terug te vinden is:

Niemand *is onsterfelijk* $\rightsquigarrow \quad \neg \exists x \in M : \neg Sx.$

Alle mensen *zijn sterfelijk* $\rightsquigarrow \quad \forall x \in M : Sx..$

Opgave 15.4 Kies een geschikt discussiedomein, geef een vertaalsleutel, en vertaal de volgende zinnen met behulp van beperkte kwantificatie:

1. Niet iedereen is aardig.
2. Sommige vrouwen minachten iedere man.
3. Als niemand iets weet verklapt niemand iets.
4. Iedereen houdt van iemand, en sommigen houden van iedereen.

De linguïstische voordelen van beperkte kwantificatie zijn echter nogal betrekkelijk. Uniek bepalende beschrijvingen (*de koningin*, *de directeur*, *de decaan van de faculteit*) zijn, als we ze Russelliaans aanpakken (vergelijk § 11.4), nog steeds wat moeilijk terug te vinden. *De koningin is jarig* wordt nu:

$$\exists x \in K : (\forall y \in K : y = x \wedge Jx).$$

Verder mogen we de ogen niet sluiten voor het feit dat we met de nu voorhanden middelen de volgende nominale constituenten nog helemaal niet kunnen analyseren:

De meeste taalkundigen *houden van goede wijn.*

De helft van de bevolking *is tegen kernenergie (maar wil wel veel en goedkope elektriciteit).*

In [Barwise & Cooper 1981] wordt bewezen dat *de meeste A zijn B* en *de helft van de A zijn B* niet in eerste orde predikatenlogica kunnen worden uitgedrukt, gesteld tenminste dat we behalve de eenplaatsige predikaatletters A en B geen predikaatletters ter beschikking hebben. Als er een extra tweeplaatsige predikaatletter R mag worden gebruikt en we ons beperken tot eindige modellen kan het weer wel: zie [Thijsse 1983]. De behandeling van uitdrukkingen als *de helft* en *de meeste* komt in § 15.3 aan de orde. Toch geldt juist voor deze hogere orde kwantoren dat ze *essentieel* beperkt zijn. "De meeste A zijn B" is bij voorbeeld niet weer te geven als "Voor de meeste dingen geldt: als ze A zijn, dan zijn ze ook B."

15.2 Tweede orde logica

De predikaten-logica zoals gepresenteerd in hoofdstuk 6 heet ook wel *eerste orde logica*. De reden: de objecten waarover in de gewone predikatenlogica wordt gekwantificeerd heten, in de terminologie van Bertrand Russell, *objecten van de eerste orde*. Russell deelde objecten in naar de interne verzamelingstheoretische structuur die ze hebben. *Objecten van de eerste orde* zijn objecten die zelf niet weer verzamelingen zijn, of liever: objecten waarvan de eventuele verzamelingstheoretische interne structuur niet door de taal in kwestie beschreven wordt. Een theelepeltje is dus een object van de eerste orde. Een *verzameling* van theelepeltjes heeft een structuur die een graadje ingewikkelder is; Russell noemt dit daarom een *object van de tweede orde*. Een verzameling die bestaat uit verzamelingen theelepeltjes en verzamelingen suikerklontjes is een *object van de derde orde*, enzovoorts.

Tweede orde logica ontstaat door ook kwantificatie toe te laten over objecten van de tweede orde. Als een domein van (eerste orde) objecten D gegeven is, dan zijn alle deelverzamelingen van D en alle twee- of meerplaatsige relaties op D objecten van de tweede orde.

De predikaatletters uit de eerste orde logica waren (eerste orde) predikaat*constanten*: namen voor predikaten (eigenschappen of relaties) die gedefinieerd zijn voor objecten van de eerste orde. Daarnaast staan we nu ook (eerste orde) predikaat*variabelen* toe. Laat $A, B, C \ldots$ eerste orde predikaatconstanten zijn en $X, Y, Z \ldots$ eerste orde predikaatvariabelen. Nu kunnen we kwantificeren over eerste orde eigenschappen; die eigenschappen worden dan zelf beschouwd als objecten van de tweede orde. Kortom: de interpretatie van een term is een object van de eerste orde, en de interpretatie van een eerste orde predikaat is een object van de tweede orde.

Jan heeft een eigenschap die Marie niet heeft
$\rightsquigarrow \quad \exists X(Xj \wedge \neg Xm).$

Jan en Marie hebben geen enkele eigenschap gemeen
$\rightsquigarrow \quad \forall X(Xj \to \neg Xm)..$

"Piet heeft alle kenmerken van een machistische man" kan worden geparafraseerd als: "Piet heeft alle kenmerken die elke machistische man heeft", dus de vertaling in tweede orde logica ziet er als volgt uit:

$$\forall X(\forall x((Mx \wedge M'x) \to Xx) \to Xp).$$

Opgave 15.5 Vertaal nu zelf: "Pinochet heeft alle kenmerken van een gevreesd dictator, en geen enkel kenmerk van een geliefd staatshoofd."

De filosoof Leibniz bedacht al in de zeventiende eeuw dat *identiteit* te definiëren valt met behulp van kwantificatie over eigenschappen van dingen. Hij stelde de volgende twee principes op:

– **Principe van de gelijkheid van wat niet te onderscheiden valt**

  $\forall x \forall y(\forall X(Xx \rightarrow Xy) \rightarrow x = y)$.

– **Principe van de ononderscheidbaarheid van wat identiek is**

  $\forall x \forall y(x = y \rightarrow \forall X(Xx \rightarrow Xy))$.

Samen leveren deze principes de volgende definitie van = op:

Definitie 15.1 [identiteit] $x = y$ *betekent* $\forall X(Xx \rightarrow Xy)$.

Dat dit mag volgt uit de combinatie van de twee principes van Leibniz: $\forall x \forall y(x = y \leftrightarrow \forall X(Xx \rightarrow Xy))$.

Opgave 15.6 Geef een parafrase van de zin "De koning toornt" in tweede orde logica, zonder gebruik te maken van het identiteitssymbool.

Opgave 15.7 In bovenstaande identiteitsprincipes ligt het misschien meer voor de hand $Xx \leftrightarrow Xy$ te gebruiken in plaats van $Xx \rightarrow Xy$. Laat evenwel zien dat $\forall X(Xx \rightarrow Xy)$ equivalent is met $\forall X(Xx \leftrightarrow Xy)$.

Nu, zul je zeggen, wanneer tweede orde predikatenlogica blijkbaar zoveel uitdrukkingskracht heeft dat we zelfs identiteit erin kunnen definiëren, waarom dan niet voor eens en voor altijd overgestapt van eerste- naar tweede orde logica? De kink in de kabel is dat we dan de mooie correspondentie tussen $\vdash$ en $\models$ die we in de eerste orde logica hadden kwijt zouden zijn: er bestaat geen 'mooie' (dat wil zeggen: recursief definieerbare) verzameling axioma's die als stellingen precies de verzameling van universeel geldige tweede orde formules oplevert.

Niettemin speelt kwantificatie over tweede orde objecten (tweede orde logica dus) in semantische theorieën over natuurlijke taal vaak een grote rol. Een belangrijke toepassing is de theorie van de zogenaamde *gegeneraliseerde kwantoren* voor het verantwoorden van de betekenis van nominale constituenten.

15.3 Gegeneraliseerde kwantoren-theorie

We hebben in 15.1 *beperkte kwantificatie* geïntroduceerd als een notatie-variant van gewone eerste orde existentiële en universele kwantificatie. Een voorbeeld van een formule met een beperkte universele kwantor is: $\forall x \in A : Bx$. Een formule met een beperkte existentiële kwantor: $\exists x \in A : Bx$. Het idee van de gegeneraliseerde kwantoren-theorie is dat zo'n beperkte kwantor-formule eigenlijk iets zegt over de manier waarop *twee verzamelingen* (twee tweede orde objecten dus) zich tot elkaar verhouden. Bij de voorbeeld-formules zijn dat, gegeven zeker model $\mathscr{M} = \langle D, I \rangle$, de verzamelingen $I(A)$ en $I(B)$.

Heel algemeen zeggen we nu: een kwantor is een relatie tussen tweede orde objecten (verzamelingen). In het geval van de universele kwantor is het de relatie *bevat zijn in*. Immers: "Alle A zijn B" is waar in model $\mathscr{M}$ desda $I(\mathrm{A}) \subseteq I(\mathrm{B})$. In het geval van de existentiële kwantor is het de relatie *een niet-lege doorsnede hebben met*. Ga maar na: "Minstens één A is B" is waar desda $I(\mathrm{A}) \cap I(\mathrm{B}) \neq \emptyset$.

Goed, de beperkte universele kwantor en de beperkte existentiële kwantor kunnen dus worden beschouwd als relaties op de machtsverzameling van het domein D van onze modellen. Formeel: de interpretatie van *alle* in model $\mathscr{M} = \langle D, I \rangle$ is de relatie

$$\{\langle A, B \rangle \mid A \subseteq B \subseteq D\}.$$

De interpretatie van *minstens één* in model $\mathscr{M} = \langle D, I \rangle$ is de relatie

$$\{\langle A, B \rangle \mid A \subseteq D \text{ en } B \subseteq D \text{ en } A \cap B \neq \emptyset\}.$$

Gegeven dit perspectief ligt het nu voor de hand om ook *andere* relaties op de machtsverzameling van D te beschouwen dan alleen deze twee. Er blijkt dan meteen dat we nu ook niet-standaard kwantificatie kunnen behandelen, zoals in "minstens twee A zijn B" en "minstens de helft van de A zijn B". Hiertoe maken we gebruik van de notatie $|A|$ voor 'het aantal elementen dat A bevat'. De interpretatie van *minstens twee* wordt nu (gegeven domein D):

$$\{\langle A, B \rangle \mid A \subseteq D \text{ en } B \subseteq D \text{ en } |A \cap B| \geq 2\}.$$

Minstens de helft wordt geïnterpreteerd als de volgende relatie op de machtsverzameling van het individuendomein D:

$$\{\langle A, B \rangle \mid A \subseteq D \text{ en } B \subseteq D \text{ en } |A - B| \leq |A \cap B|\}.$$

Opgave 15.8 Welke relatie op $2^{(D)}$ kan nu dienen als de interpretatie van 'hoogstens vijf'?

Opgave 15.9 Welke relatie op $2^{(D)}$ kan dienen als interpretatie van 'geen'?

Opgave 15.10 Welke relatie op $2^{(D)}$ kan dienen als interpretatie van 'niet alle'?

Je ziet aan de opdrachten: we hebben hier een theorie te pakken die ons de middelen in handen geeft om—heel algemeen—de interpretatie van de lidwoordgroep in nominale constituenten ter hand te nemen. Dankzij het gegeneraliseerde kwantoren-perspectief is het blikveld niet langer beperkt tot existentiële en universele kwantificatie, maar komen ook andere kwantoren aan bod. Een lidwoord-groep als *geen*, *minstens twee*, *niet alle*, *precies drie*, *hoogstens vier* wordt wel een *determinator* genoemd (Eng.: *determiner*); diezelfde term is trouwens ook in gebruik voor de relatie op $2^{(D)}$ die dient als *interpretatie* voor zo'n lidwoordgroep. Een kwantificerende determinator wordt een *kwantor* genoemd.

Is het nu zo dat *elke* relatie op $2^{(D)}$ de interpretatie is van een kwantor? Dat ligt er natuurlijk maar aan wat we willen verstaan onder een kwantor, maar het ligt voor de hand om een paar beperkingen op te leggen. Daartoe zullen we een tweetal *eigenschappen* van determinatoren (determinator-interpretaties) gaan isoleren.

In de eerste plaats willen we de intuïtie uitdrukken dat zinnen als *Alle jongens wandelen*, *Minstens twee jongens hollen*, *Hoogstens drie jongens praten* betrekking hebben op de verzameling *jongens* in het discussiedomein. Alleen het gedrag van de jongens in het domein maakt iets uit voor de waarheidswaarde van deze zinnen. Of de andere individuen in het domein wandelen, lopen of praten doet er niet toe. We hoeven dus, om te kijken of deze zinnen waar zijn, niet *alle* wandelaars, hardlopers of praters te onderzoeken, maar alleen de wandelende jongens, de hollende jongens, of de pratende jongens. Nog anders gezegd: de volgende parafrases veranderen de betekenis van de bovenstaande zinnen niet: *Alle jongens zijn wandelende jongens*; *Minstens twee jongens zijn hollende jongens*; *Hoogstens drie jongens zijn pratende jongens*. Merk verder op dat het voor de waarheidswaarde van deze zinnen niet uitmaakt of we het discussiedomein groter maken of kleiner, zolang we de verzameling jongens maar intact laten. Het is de verzameling jongens die bepalend is voor de vraag of de kwantorrelatie al dan niet geldt. Algemeen kunnen we dus zeggen:

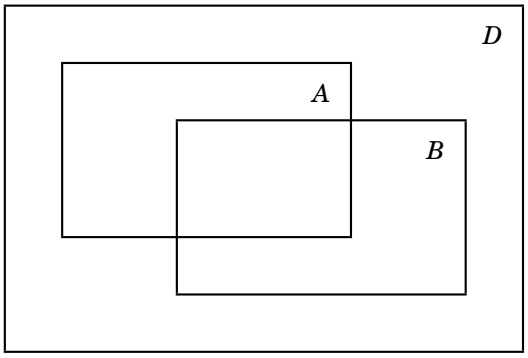
> Als een kwantor-relatie op de machtsverzameling van het domein D geldt tussen een verzameling A en een verzameling B, dan geldt die relatie ook op de machtsverzameling van het *subdomein* A tussen A en de *doorsnede* van A en B, en vice versa.

Deze eigenschap noemen we de *conservativiteitseigenschap* van de relatie. We gebruiken R_D voor een tweeplaatsige relatie op $2^{(D)}$. We kunnen als volgt uitdrukken dat R_D conservatief is:

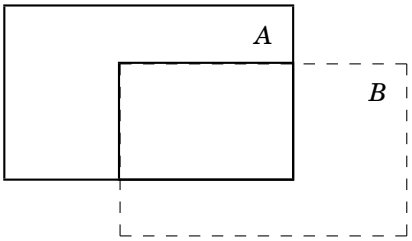
Definitie 15.2 Conservativiteit van R_D

Voor alle $A,B \subseteq D$: R_DAB desda $R_AA(A \cap B)$.

Om te zien wat *conservativiteit* precies uitdrukt maken we een plaatje van een domein D, met twee verzamelingen A en B.



Stel dat een kwantor-relatie op $2^{(D)}$ geldt tussen A en B. Dan zegt *conservativiteit* dat die relatie ook geldt tussen A en $A \cap B$, als we het domein van de kwantor-relatie beperken tot $2^{(A)}$, en vice versa. Dit betekent dat we het bovenstaande plaatje zonder bezwaar mogen vervangen door het volgende plaatje:



Denk bij voorbeeld bij A aan de verzameling jongens, bij B aan de verzameling praters. Dan is $A \cap B$ de verzameling pratende jongens. Buiten de verzameling jongens, het eerste argument van de relatie, hoeven we niet te kijken. We zeggen dat een kwantor-relatie *teert op* zijn eerste argument. In het bovenstaande voorbeeld: de verzameling A (de verzameling jongens). De conservativiteits-eigenschap wordt ook wel de *teren-op eigenschap* (Eng.: *the live-on property*) genoemd.

Uit de conservativiteits-eigenschap volgt meteen dat we de index D van een kwantor-relatie R_D gerust mogen weglaten. Voor de interpretatie doet het gedeelte van het domein dat ligt buiten de verzameling waarop de kwantor-relatie teert er immers niet toe.

Een voor de hand liggende tweede eis die we aan kwantor-relaties kunnen stellen is dat ze inderdaad iets te maken hebben met kwantiteit. De conservativiteits-eis zegt ons dat het antwoord op de vraag of R geldt tussen A en B alleen mag afhangen van de verzamelingen A en $A \cap B$. De kwantiteits-eis voegt daar nog iets aan toe: het gaat niet om deze verzamelingen zonder meer, maar alleen om de *aantallen elementen* die ze bevatten. “Alle jongens wandelen” is waar precies wanneer het *aantal* jongens dat niet wandelt 0 is; “Hoogstens drie jongens praten” is waar wanneer het *aantal* jongens dat praat ≤ 3 is, enzovoorts. De kwantiteits-eis op kwantor-relaties wordt nu:

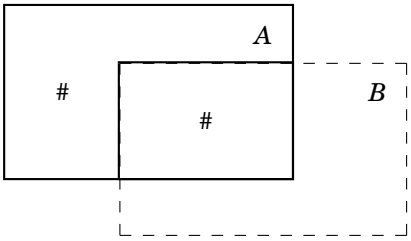
> Als de kwantor-relatie R geldt tussen twee verzamelingen A en B, en we hebben twee andere verzamelingen C en D waarvan we weten dat C evenveel elementen heeft als A, en de doorsnede van C en D evenveel elementen als de doorsnede van A en B, dan weten we dat de kwantor-relatie ook moet gelden tussen C en D.

Controleer dit zelf aan de hand van concrete voorbeelden als “Alle jongens wandelen”, “Minstens twee jongens hollen”, “Hoogstens drie jongens praten”, “De helft van de jongens voetbalt”. Iets formeler kunnen we de kwantiteits-eigenschap als volgt formuleren:

Definitie 15.3 Kwantiteits-eigenschap van R

Voor alle X,Y,Z,U:
als $R(X,Y)$ en $|X| = |Z|$ en $|X \cap Y| = |Z \cap U|$, dan $R(Z,U)$.

Wanneer we nu weer een plaatje gaan tekenen van twee verzamelingen A en B waartussen een kwantor-relatie geldt, dan zien we dat we ons kunnen beperken tot het aangeven van twee kardinaalgetallen, (wanneer we ons beperken tot eindige domeinen: twee natuurlijke getallen). Het eerste getal geeft het aantal elementen in $A - B$ aan, het tweede getal het aantal elementen in $A \cap B$:

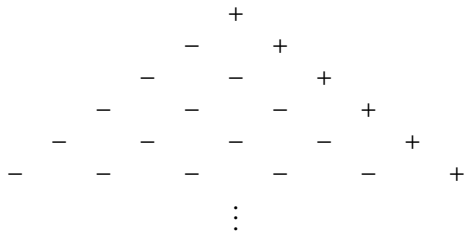


De twee #-tekens in het plaatje geven de getallen $|A - B|$ en $|A \cap B|$ aan. We hadden natuurlijk ook de twee getallen $|A|$ en $|A \cap B|$ kunnen nemen: als we $|A|$ hebben weten we ook $|A - B|$ en omgekeerd, vanwege de betrekking $|A - B| = |A| - |A \cap B|$ (we gaan er nu even van uit dat de verzamelingen A en B eindig zijn).

Aangenomen dat kwantor-relaties voldoen aan *conservativiteit* en *kwantiteit* kunnen we *elke* kwantor-relatie karakteriseren als een patroon van plaatsen in een oneindige getallen-piramide. Ook wanneer we ons beperken tot eindige domeinen is de getallen-piramide oneindig: er is immers geen bovengrens aan de grootte van onze domeinen. Hier volgt het recept voor het construeren van de getallen-piramides. Elke laag van de piramide behandelt de verdeling van $|A - B|$ en $|A \cap B|$, voor een bepaald aantal $|A|$. De top van de piramide bestaat uit het ene getallenpaar $\langle 0,0 \rangle$, want als er 0 A's zijn is dit de enig mogelijke verdeling. De top vormde de eerste laag. De tweede laag behandelt het geval waar er 1 A is; zij bestaat uit de getallenparen $\langle 1,0 \rangle$ ("er is 1 ding dat A is maar niet B, en er zijn geen dingen die zowel A als B zijn") en $\langle 0,1 \rangle$ ("er zijn geen dingen die alleen A zijn maar niet B, en er is 1 ding dat zowel A als B is"). Net zo voor de volgende lagen. De hele piramide ziet er dus uit als in Figuur 15.1.

Heel fraai is, dat we nu kwantor-relaties kunnen karakteriseren met behulp van een *patroon* van + en − dat ze in deze piramide teweeg brengen: een + verschijnt op de plaatsen van de getallenparen die de aantallen $|A - B|$ en $|A \cap B|$ aangeven van de verzamelingen A en B waarvoor de kwantor-relatie geldt, een − op alle andere plaatsen. Het volgende voorbeeld maakt dit hopelijk duidelijk. "Alle A zijn B" is immers waar als $A \subseteq B$ (eigenlijk: als $I(A) \subseteq I(B)$, maar hierover doen we even niet moeilijk). Maar $A \subseteq B$ is precies het geval als $A - B = \emptyset$, en dit is weer equivalent met $|A - B| = 0$.

alle A zijn B

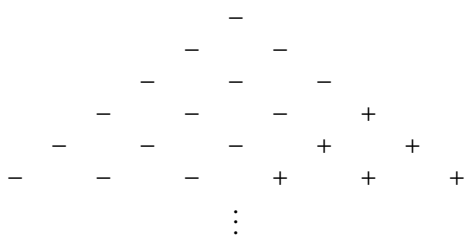


Opgave 15.11

1. Geef het piramide-patroon voor *geen*.
2. Geef het piramide-patroon voor *niet alle*.
3. Geef het piramide-patroon voor *minstens één*.
4. Hoe verhouden deze piramide-patronen zich tot elkaar en tot het patroon voor *alle* ?

Hier is het piramide-patroon voor *minstens drie*:

minstens drie A zijn B



Opgave 15.12

1. Geef het piramide-patroon voor *hoogstens drie*.
2. Hoe verhoudt dit patroon zich tot dat van *minstens drie*?
3. Geef ook het piramide patroon voor *precies drie*?
4. Hoe verhoudt dit patroon zich tot die van *hoogstens drie* en *minstens drie*?
5. Verklaar het gevonden verband.

Opgave 15.13

1. Geef het piramide-patroon voor *minstens twee en hoogstens vijf*.
2. Geef het piramide-patroon voor *precies twee of minstens vier*.
3. Geef het piramide-patroon voor *precies twee of precies vijf*.
4. Tenslotte dat voor *minstens de helft*.

In het gegeneraliseerde kwantor-perspectief zien kwantor-interpretaties eruit als relaties. We kunnen kwantoren dus gaan klassificeren met behulp van relatie-eigenschappen (vergelijk § 3.1). Enkele voorbeelden. De determinator *alle* is reflexief. Immers, de volgende beweringen zijn altijd waar: *alle wandelaars wandelen*, *alle jongens zijn jongens*, *alle meisjes zijn meisjes*, enzovoort. Dus in het algemeen: als 𝒬

$\|A\| = 0$								$0,0$								
$\|A\| = 1$							$1,0$			$0,1$						
$\|A\| = 2$					$2,0$			$1,1$			$0,2$					
$\|A\| = 3$			$3,0$			$2,1$			$1,2$			$0,3$				
$\|A\| = 4$		$4,0$			$3,1$			$2,2$			$1,3$			$0,4$		
$\|A\| = 5$	$5,0$			$4,1$			$3,2$			$2,3$			$1,4$			$0,5$
$\vdots$								$\vdots$								

Figuur 15.1: Getallen-piramide.

geïnterpreteerd wordt als *alle* (dat wil zeggen: als de inclusie-relatie), dan hebben we:

$$\forall X \mathcal{Q} X X.$$

Dit drukt de reflexiviteits-eigenschap van $\mathcal{Q}$ uit.

De determinator *alle* is transitief. Immers, uit *alle jongens voetballen* en *alle voetballers hollen* volgt: *alle jongens hollen*. Uit *alle meisjes huilen* en *alle huilenden treuren* volgt: *alle meisjes treuren*, enzovoort. In het algemeen: als $\mathcal{Q}$ geïnterpreteerd wordt als *alle*, dan hebben we:

$$\forall X \forall Y \forall Z((\mathcal{Q} X Y \wedge \mathcal{Q} Y Z) \rightarrow \mathcal{Q} X Z).$$

Dit drukt de transitiviteits-eigenschap van $\mathcal{Q}$ uit.

De determinator *alle* is antisymmetrisch. Uit *alle jongens voetballen* en *alle voetballers zijn jongens* volgt dat de verzameling voetballers identiek is aan de verzameling jongens. Algemeen: als Q geïnterpreteerd wordt als *alle*, dan hebben we:

$$\forall X \forall Y((\mathcal{Q} X Y \wedge \mathcal{Q} Y X) \rightarrow X = Y).$$

Deze formule drukt uit dat de relatie $\mathcal{Q}$ anti-symmetrisch is.

Opgave 15.14

1. Is de determinator *geen* reflexief?
2. Irreflexief?
3. Symmetrisch?
4. Antisymmetrisch?
5. Transitief?

Opgave 15.15 Dezelfde vragen voor de determinator *precies twee*.

Opgave 15.16

1. Zij gegeven dat de determinator $\mathcal{Q}$ reflexief is. Wat kun je hieruit opmaken voor het piramide-patroon van deze determinator?
2. Laat zien dat elke determinator met het gevonden piramide-patroon reflexief is.

Wanneer je de laatste opdracht heeft begrepen en uitgevoerd kun je voor alle determinatoren waarvoor je een piramide-patroon heeft getekend in één oogopslag zien of ze reflexief zijn of niet.

Opgave 15.17 Zij gegeven dat het piramide-patroon van de determinator $\mathcal{Q}$ aan de rechter-zijkant louter minnen vertoont. Welke relationele eigenschap voor de interpretatie van $\mathcal{Q}$ kun je hieruit afleiden?

Wanneer bovenstaande opdrachten je nieuwsgierig hebben gemaakt, dan is er werk aan de winkel voor je. Vele fraaie vraagstukken in dit genre (zij het soms wat ingewikkelder) liggen nog op je te wachten in de gegeneraliseerde kwantoren theorie.

De theorie der gegeneraliseerde kwantoren speelt impliciet een belangrijke rol in de semantiek van de Montague grammatica. Taalkundig is de theorie van belang omdat zij ons in staat stelt allerlei taalkundige kwantificatieverschijnselen, bij voorbeeld in gekwantificeerde lidwoordgroepen als *sommige*, *de meeste*, *vele*, ..., maar ook in temporele adverbia als *soms*, *meestal*, *vaak*, ..., logisch onder één hoedje te vangen.

De theorie heeft door het verschijnen van [Barwise & Cooper 1981] een nieuwe impuls gekregen. Aan de hoge vlucht die de theorie daarna heeft genomen hebben Nederlandse logici en semantici een belangrijke bijdrage geleverd. Zie voor meer informatie: [Van Benthem & Ter Meulen 1985], en de daar genoemde literatuur. Taalkundige toepassingen zijn te vinden in [Zwarts 1981, Zwarts 1986]. Om het taalkundige belang van gegeneraliseerde kwantorentheorie te illustreren definiëren we een nieuwe eigenschap van kwantoren.

Definitie 15.4 Opwaartse monotonie rechts van een kwantor

Voor alle X, Y, Y': als $\mathcal{Q} X Y$ en $Y \subseteq Y'$ dan $\mathcal{Q} X Y'$.

Er bestaat een tegenhanger voor opwaartse monotonie:

Definitie 15.5 Neerwaartse monotonie rechts van een kwantor

Voor alle X, Y, Y': als $\mathcal{Q} X Y$ en $Y' \subseteq Y$ dan $\mathcal{Q} X Y'$.

Voorbeelden van kwantoren die opwaarts monotoon zijn (of kortweg stijgend) in hun rechterargument zijn *alle* en *minstens één*. Dat dit zo is blijkt uit het feit dat de volgende redeneringen geldig zijn:

Alle jongens zijn aardig.
Alle jongens zijn aardig of verlegen.

Minstens één meisje is aardig.
Minstens één meisje is mooi of aardig.

Voorbeelden van kwantoren die neerwaarts monotoon zijn (of kortweg dalend) in hun rechterargument zijn *geen* en *weinig*.

Opgave 15.18 Geef voorbeelden waaruit dit blijkt.

De monotonie eigenschappen van kwantoren zijn zeer nuttig voor het beschrijven van zogenaamde *polariteitsverschijnselen* in natuurlijke taal. Het Nederlandse werkwoord *hoeven* is een voorbeeld van een werkwoord dat alleen kan worden gebruikt in een 'negatieve context'. In taalkundig jargon: het is een woord met *negatieve polariteit*. Beschouw nu de volgende voorbeelden:

Hoogstens honderd strijders hoeven de wacht te houden.

**Minstens honderd strijders hoeven de wacht te houden.*

Ga na dat de kwantor *hoogstens honderd* neerwaarts monotoon is in zijn rechterargument. De kwantor *minstens honderd* is dat niet. De nominale constituenten die bij combinatie met een verbale constituent een woord met negatieve polariteit in die verbale constituent toelaten blijken precies de kwantoren te zijn die neerwaarts monotoon zijn in hun tweede argument. Het begrip *neerwaartse monotonie* levert een bruikbare precisering op van het veel vagere begrip 'negatieve context'.

Opgave 15.19 Beschouw de volgende voorbeelden:

– Alle mannen waren allerminst tevreden.

– Precies vijf mannen waren allerminst tevreden.

– *Niemand was allerminst tevreden.

Het Nederlandse woord *allerminst* is een woord met *positieve polariteit*. Formuleer in termen van monotonie een principe waaraan het gebruik van dit woord moet voldoen.

Opgave 15.20 Formuleer twee monotonie principes voor het linkerargument van een kwantor. Beschouw nu de volgende kwantoren:

1. alle
2. niet alle
3. minstens een
4. hoogstens honderd
5. precies vijf
6. geen.

Ga na wat hun monotonie-eigenschappen zijn in het linkerargument.

15.4 Extensionele typenlogica

In 15.2 hebben we variabelen voor eerste orde predikaten (en kwantificatie over die variabelen) ingevoerd. Wanneer we ook nog constanten voor tweede orde predikaten toestaan kunnen we predikatie over eerste orde predikaten uitdrukken, en bij voorbeeld een vertaling geven voor "Liefde is blind". Namen voor tweede orde predikaten hebben we ook nodig in de vertaling van "Jan heeft alleen goede eigenschappen". Dit wordt:

$$\forall X(Xj \rightarrow \mathbf{G}X),$$

waarbij X een eerste orde predikaatvariabele is, en $\mathbf{G}$ een tweede orde predikaatconstante. "Ieder mens heeft wel een goede eigenschap" wordt nu:

$$\forall z(Mz \rightarrow \exists X(Xz \wedge \mathbf{G}X)).$$

Voortgaande op het pad dat we in § 15.2 zijn ingeslagen zouden we ook kunnen gaan kwantificeren over predikaten van predikaten. Zo ontstaat derde orde logica. Om systematische redenen is het nu aantrekkelijk om een *algemene* theorie te ontwikkelen die kwantificatie over steeds ingewikkelder objecten toestaat.

Zo'n theorie is gepresenteerd door Russell. Het is de zogenaamde typenlogica, waarin kwantificatie over alle eindige ordes is toegestaan. Russell had zijn theorie bedacht om de door hem ontdekte paradox in de verzamelingenleer (vergelijk § 4.8) te omzeilen, maar in dat verband is ze eigenlijk nooit populair geworden: het bleek dat je die paradox (en verwante paradoxen) ook al kwijt kon raken door het kiezen van geschikte axioma's in de taal van de eerste orde predikatenlogica (zie § 14.2).

De typenlogica is echter wel zeer geschikt voor het analyseren van natuurlijke taal. Het blijkt een zeer natuurlijke pendant te zijn van de zogenaamde *categoriale syntaxis*. We zullen hier iets vertellen over typenlogica in zijn allersimpelste vorm, waarin alle typen opgebouwd zijn uit de basistypen *objecten* en *waarheidswaarden*. Deze versie van de typenlogica heet *extensionele typenlogica*.

In de extensionele typenlogica is het mogelijk te verwijzen naar predikaten van willekeurige complexiteit (dat wil zeggen: van een willekeurige eindige orde, of: van een willekeurig *type*). Dat dit voor de analyse van natuurlijke taal handig is, is duidelijk te zien in voorbeelden waarin de predikaten waarvan iets gezegd wordt zelf ook nog eens *interne structuur* vertonen:

Niet roddelen *is verstandig.*

Iemand bestelen *is strafbaar.*

Opstaan voor iemand *misstaat niemand.*

Zijn naasten liefhebben *is ieders plicht.*

In de typenlogica wordt de verwijzing naar een predikaat tot stand gebracht door middel van zogenaamde *lambda-abstractie* (of: λ-abstractie). Laat een uitdrukking φ waarin een constante a voorkomt gegeven zijn. We noteren dit als $\varphi(a)$. Deze uitdrukking valt te beschouwen als een zin waarin een predikaat aan a wordt toegekend. Dat predikaat willen we er nu uit peuteren door middel van λ-abstractie. Daartoe vervangen we overal in $\varphi(a)$ de constante a door de variabele x. (We nemen aan dat x een variabele is die niet vrij in $\varphi(a)$ voorkomt.) Dit kunnen we schrijven als $[x/a]\varphi(a)$ of als $\varphi(x)$. Vervolgens markeren we de variabele x met een zogenaamde λ-operator. Het resultaat is: $\lambda x[\varphi(x)]$. De vierkante haken geven hier het bereik van de λ-operator aan. Neem even aan dat $\varphi(a)$ een *zin* was (een gesloten formule). Die zin drukt uit dat object a zekere eigenschap heeft. Nu staat $\lambda x[\varphi(x)]$ voor een karakteristieke functie, namelijk de *functie* die objecten d afbeeldt op 1 of op 0, al naar gelang ze al of niet de eigenschap hebben die $\varphi(a)$ aan a toeschreef. Enkele concrete voorbeelden kunnen dit duidelijker maken.

- Abstractie op a in Pa levert $\lambda x[Px]$ ('de eigenschap P hebben').
- Abstractie op a in Rab levert $\lambda x[Rxb]$ ('in relatie R staan tot b', 'b R-en').
- Abstractie op b in Rab levert $\lambda x[Rax]$ ('door a ge-R-d worden').
- Abstractie op a in Raa levert: $\lambda x[Rxx]$ ('tot zichzelf in relatie R staan'; 'zichzelf R-en').
- Abstractie op a in $\neg Pa$ levert: $\lambda x[\neg Px]$ ('de eigenschap niet-P hebben').

Met behulp van de λ-operator kunnen we nu de hierboven gegeven voorbeelden uit de natuurlijke taal gaan vertalen. We maken weer gebruik van hogere orde predikaatconstanten, waarvoor we vette letters zullen nemen. "Niet roddelen is verstandig" krijgt als vertaling: $\mathbf{V}(\lambda x[\neg Rx])$. De vertaling van "Iemand bestelen is strafbaar" kan nu worden: $\mathbf{S}(\lambda x[\exists y Bxy])$. "Opstaan voor iemand misstaat niemand": $\neg\exists z\mathbf{M}(\lambda x[\exists y Oxy], z)$. Toelichting: $\mathbf{M}$ is een hogere orde predikaatconstante die als *eerste* argument een eerste orde predikaat neemt en als *tweede* argument een object. $\mathbf{M}$ staat dus voor: *eigenschap* zus hebben misstaat aan *object* zo. Tenslotte de vertaling voor "(Al) zijn naasten liefhebben is ieders plicht": $\forall z\mathbf{P}(\lambda x[\forall y(Nyx \rightarrow Lxy)], z)$. Toelichting: $\mathbf{P}$ staat voor: *eigenschap* zus hebben is plicht voor *object* zo.

Het bovenstaande was niet meer dan een eerste stap. Behalve λ-abstractie uit zinnen is ook abstractie uit predikaten mogelijk. Abstractie op a in $\lambda x[Rxa]$ levert $\lambda y[\lambda x[Rxy]]$, hetgeen uitdrukt: '$R$-en', ofwel: 'in relatie R staan'. We kunnen ook abstractie plegen op a in $\lambda x[Rax]$. Dit levert de uitdrukking $\lambda y[\lambda x[Ryx]]$, met een andere betekenis: 'ge-R-d worden'. Over dit verschil tussen de 'actieve' en de 'passieve' vorm van een relatie-uitdrukking volgt straks meer informatie.

Tot nu toe hebben we de λ-operator alleen toegepast op *individuele constanten*. We kunnen ook abstractie toepassen op predikaatconstanten. Abstractie op A in Ab (A is een predikaatconstante) levert op: $\lambda Y[Yb]$. In deze uitdrukking is Y een predikaatvariabele. De betekenis van de hele uitdrukking: 'eigenschap zijn van b'.

In de analyse van natuurlijke taal kunnen we nu dit alles mooi gebruiken. Gegeven dat we de bijdrage van de afzonderlijke woorden aan de betekenis van een zin als "Jan loopt snel" willen achterhalen, dan kan dat met behulp van vertaling in een typenlogische taal als volgt:

- "Jan loopt" wordt vertaald als Lj.

Hieruit volgt meteen:

- 'Jan' wordt vertaald als j, en 'loopt' als L, of met gebruik van λ-notatie, als $\lambda x[Lx]$.
- "Jan loopt snel" betekent zo ongeveer: "Jan heeft de eigenschap snel te lopen", waarbij *snel lopen* een eigenschap is die is afgeleid van *lopen*. De vertaling kan dus zoiets worden als $\mathscr{S}Lj$, waarbij $\mathscr{S}$ een uitdrukking is die van een eenplaatsig eerste orde predikaat een nieuw eenplaatsig eerste orde predikaat maakt.

Korte discussie tussendoor: deze vertaling lijkt misschien nodeloos ingewikkeld, want we zouden immers ook kunnen vertalen in $Lj \wedge Sj$, dat wil zeggen: Jan loopt en hij *is* snel. Dit is echter niet helemaal correct. Immers, stel dat Jan zich te voet met een snelheid van twaalf kilometer per uur voortspoedt en daarbij op gedragen toon een gedicht van Vondel reciteert. Nu is "Jan loopt snel" zeker waar; ook "Jan reciteert" is waar; maar: "Jan reciteert snel" is niet waar. Het voorbeeld geeft aan dat de parafrase van de eerste zin zoiets zou moeten zijn als "Jan loopt, en dat lopen heeft de eigenschap snel te zijn (vergeleken bij andere loopactiviteiten)". Een andere mogelijkheid is: $Lj \wedge \mathbf{S}L$ (waarbij $\mathbf{S}$ een uitdrukking is die met een eenplaatsig eerste orde predikaat combineert tot een zin. Dit is echter ook niet helemaal juist. Immers, gegeven dit recept zou "Jan loopt snel, en Elias loopt niet snel" als vertaling hebben $Lj \wedge \mathbf{S}L \wedge Le \wedge \neg\mathbf{S}L$. Maar dit is in geen enkele 'redelijke' situatie waar, zodat deze vertaling niet correct kan zijn. De gekozen vertaling lost allebei deze problemen op. Hieruit volgt:

- 'loopt snel' moet als vertaling hebben: $\lambda x[\mathscr{S}Lx]$.

Maar als we dit hebben weten we ook hoe we 'snel' moeten vertalen:

- 'snel' heeft als vertaling: $\lambda Y[\lambda x[\mathscr{S}Yx]]$.

Dit voorbeeld was bedoeld om te illustreren dat de uitdrukkingskracht van een typen-theoretische taal uitstekend van pas komt wanneer het erom gaat de betekenisdragende bouwstenen van de natuurlijke taal elk aan een eigen vertaling te helpen.

Syntactisch gezien is het bijwoord 'snel' een zinsdeel dat samen met de verbale constituent 'loopt' een nieuwe verbale constituent 'loopt snel' kan vormen. De gang van zaken op het niveau van de typenlogische vertaling vertoont nu een volledige parallel: de vertaling van 'loopt', $\lambda y[Ly]$, een eerste orde predikaat, combineert met de

vertaling van 'snel', $\lambda Y[\lambda x[\mathscr{S}Yx]]$, tot een *nieuw* eerste orde predikaat dat de vertaling vormt van 'loopt snel': $\lambda x[\mathscr{S}Lx]$.

Hoe kunnen we nu $\lambda Y[\lambda x[\mathscr{S}Yx]]$ met $\lambda y[Ly]$ combineren, zo dat het resultaat $\lambda x[\mathscr{S}Lx]$ is? Dat zit zo. De uitdrukking $\lambda Y[\lambda x[\mathscr{S}Yx]]$ staat voor een functie die eenplaatsige eerste orde eigenschappen als argument neemt, en nieuwe eenplaatsige eerste orde eigenschappen als waarde aflevert. Een eenplaatsige eerste orde eigenschap is in wezen niets anders dan een verzameling individuen (de verzameling lopers, de verzameling jongens, de verzameling snelle lopers, de verzameling snelle jongens, enzovoorts).

De uitdrukking $\lambda y[Ly]$ staat voor een eerste orde eigenschap. Net zoals we de eenplaatsige predikaatletter L kunnen combineren met de individuele constante j tot de formule Lj, kunnen we nu de volgende combinatie maken:

$$\lambda Y[\lambda x[\mathscr{S}Yx]](\lambda y[Ly]).$$

Deze uitdrukking staat voor: de *toepassing* van het tweede orde predikaat $\lambda Y[\lambda x[\mathscr{S}Yx]]$ op het argument $\lambda y[Ly]$.

De typenlogische uitdrukking $\lambda Y[\lambda x[\mathscr{S}Yx]](\lambda y[Ly])$ kan worden vereenvoudigd door middel van zogenaamde λ-*conversie*. Zij gegeven een λ-uitdrukking $\lambda v[\varphi(v)]$. Laat α een argument zijn voor die uitdrukking. α is een argument voor $\lambda v[\varphi(v)]$ wanneer α en v van hetzelfde type zijn, dat wil zeggen ze staan allebei voor een object, allebei voor een eerste orde predikaat van een bepaalde plaatsigheid, allebei voor een tweede orde predikaat van een bepaalde plaatsigheid, enzovoort. Als α een argument is voor $\lambda v[\varphi(v)]$ mogen we $\lambda v[\varphi(v)](\alpha)$ converteren tot $[\alpha/v]\varphi$, het resultaat van het vervangen van de variabele v in φ door α, overal waar deze variabele voorkomt, terwijl het voorvoegsel λv wegvalt. Voor het gemak schrijven we dit als $\varphi(\alpha)$.

Om duidelijk te maken hoe λ-conversie precies in zijn werk gaat zijn voorbeelden weer erg nuttig:

– $\lambda x[Lx](j)$ wordt geconverteerd tot Lj.

Opmerkingen: aan dit voorbeeld kun je zien dat L en $\lambda x[Lx]$ op hetzelfde neerkomen; verder mag j een argument zijn bij $\lambda x[Lx]$ omdat x en j van hetzelfde type zijn.

– $\lambda Y[Ya](P)$ wordt geconverteerd tot Pa.

Opmerking: Y en P zijn weer van hetzelfde type (beide letters staan voor eerste orde predikaten); als de twee letters uitdrukkingen van verschillende type zouden zijn geweest zou P geen argument kunnen zijn bij $\lambda Y[Ya]$.

– $\lambda Y[Ya \wedge Yb](P)$ wordt geconverteerd tot $Pa \wedge Pb$.

– $\lambda x[\neg Rxx](a)$ wordt geconverteerd tot $\neg Raa$.

Het argument bij een λ-uitdrukking kan ook een variabele zijn:

– $\lambda x[Px](y)$ wordt geconverteerd tot $\neg Py$.

– $\lambda x[\neg Px](x)$ wordt geconverteerd tot: $\neg Px$.

Moeilijkheden kunnen ontstaan wanneer een argument-variabele bij conversie gebonden dreigt te raken, door een kwantor of door een λ-operator. De argument variabele was dan niet vrij voor de variabele v waarvoor conversie plaatsvindt. Een voorbeeld hebben we in: $\lambda x[\lambda y[Rxy]](y)$. Conversie zou hier een uitdrukking opleveren waarin de argument-variabele y ten onrechte gebonden wordt, namelijk de uitdrukking $\lambda y[Ryy]$. De remedie is standaard: toepassen van λ-conversie op de uitdrukking $\lambda v[\varphi(v)](\alpha)$ is verboden als het argument α een variabele is die niet vrij is voor v in $\varphi(v)$ of variabelen *bevat* die niet vrij zijn voor v in $\varphi(v)$. In dit soort gevallen moet eerst worden overgestapt naar een alfabetische variant van de oorspronkelijke uitdrukking waarin dit probleem niet optreedt. Voorbeeld: $\lambda x[\lambda y[Rxy]](y)$ wordt eerst vervangen door de alfabetische variant $\lambda x[\lambda z[Rxz]](y)$; λ-conversie levert nu geen problemen op en geeft als resultaat $\lambda z[Ryz]$.

Het conversie-recept toepassen op $\lambda Y[\lambda x[\mathscr{S}Yx]](\lambda x[Lx])$ geeft—in twee conversie-stappen—het beoogde resultaat. De eerste conversie-stap levert op: $\lambda x[\mathscr{S}\lambda x[Lx](x)]$. In de volgende stap krijgen we via conversie van een deel-uitdrukking het uiteindelijke conversie-resultaat: $\lambda x[\mathscr{S}Lx]$. Dit is inderdaad de vertaling voor 'snel lopen' die we wilden hebben. Komt deze λ-goochelarij je nog wat vreemd voor? Niet getreurd, want oefening baart kunst:

Opgave 15.21 Vereenvoudig de volgende formules zoveel mogelijk, met behulp van λ-conversie:

1. $\lambda Y[\lambda x[Yx]](P)$
2. $\lambda Y[\lambda x[Yx]](P)(j)$
3. $\lambda Y[\lambda Z[\exists x(Yx \wedge Zx)]](P)$
4. $\lambda Y[\lambda Z[\exists x(Yx \wedge Zx)]](Y)$
5. $\lambda Y[\lambda Z[\exists x(Yx \wedge Zx)]](Z)$
6. $\lambda x[\lambda Y[Yx]](j)$
7. $\lambda Y[Yj](\lambda x[Rax])$
8. $\lambda \mathbf{X}[\mathbf{X}(\lambda y[Py])](\lambda Y[Ya])$
9. $\lambda \mathbf{Z}[\mathbf{Z}(\lambda y[Py \wedge Qy]) \vee \mathbf{Z}(\lambda x[Bx])](\lambda X[Xb])$.

Opgave 15.22 Vertaal de volgende zinnen in de extensionele typenlogica:

1. Het is verboden zich op het gras te bevinden.
   (Vertaalsleutel: g: het gras; Bxy: x bevindt zich op y; $\mathbf{V}(Y)$: Y is verboden.)
2. Het in iets geloven geeft troost.
   (Vertaalsleutel: Gxy: x gelooft in y; $\mathbf{T}(Y)$: Y geeft troost.)
3. Niemand vertrouwen is ziekelijk.
   (Vertaalsleutel: Vxy: x vertrouwt y; $\mathbf{Z}(Y)$: Y is ziekelijk.)

4. Jezelf niet vertrouwen is ziekelijk.
   (Vertaalsleutel als boven.)

5. Niet iedereen vertrouwen is niet ziekelijk.
   (Vertaalsleutel als boven.)

Opgave 15.23 Neem aan dat $Gabc$ de vertaling is van "*A*driaan geeft de *B*akoenin-biografie aan *C*ornelis". Parafraseer nu de volgende uitdrukkingen in het Nederlands:

1. $\lambda x[Gxbc]$

2. $\lambda x[Gaxc]$

3. $\lambda x[Gabx]$.

Opgave 15.24 Vertaal de volgende Nederlandse uitdrukkingen in uitdrukkingen van de typenlogica (zie vorige opdracht voor de vertaalsleutel):

1. Aan iemand de Bakoenin-biografie geven.

2. Van iemand de Bakoenin-biografie krijgen.

3. Iets aan Cornelis geven.

4. Iets van Adriaan krijgen.

5. Aan iedereen iets geven.

Wie 'iets wil bereiken' in de semantiek van de natuurlijke taal doet er goed aan zich grondig vertrouwd te maken met de λ-rekenarij (ook wel 'λ-calculus' genoemd). Befaamde linguïsten gingen je reeds voor. Een gevleugeld woord van Barbara Partee, één van hen: "λ has changed my life".

Opgave 15.25 Vertaal in een typenlogische uitdrukking: "Iets willen bereiken is prijzenswaardig."

15.5 Syntaxis en semantiek van de extensionele typenlogica

We zijn tot nu toe wat slordig geweest in het praten over de typen van de verschillende uitdrukkingen in de typenlogica. Let op: de *typen* waar we het nu over hebben zijn *functionele typen*; iets heel anders dan de *soort-typen* die in 15.1 ter sprake zijn geweest. De soort-typen uit 15.1 waren allemaal van het functionele type *entiteit* (dat wil zeggen: individueel object).

Praten over functionele typen kan heel precies. De verzameling van alle typen noemen we TYPE. Deze verzameling wordt als volgt recursief gedefinieerd:

Definitie 15.6 *Definitie van* TYPE.

- *e en t zijn elementen van* TYPE *(e staat voor 'entiteit', Eng.: entity; t staat voor 'waarheidswaarde', Eng.: truthvalue);*
- *als a en b elementen zijn van* TYPE*, dan is* $\langle a,b\rangle$ *ook een element van* TYPE*;*
- *niets anders is een element van* TYPE.

Uitdrukkingen van type $\langle a,b\rangle$, voor zekere typen a en b, staan voor functies: als je zo'n uitdrukking combineert met een uitdrukking van type a is het resultaat een uitdrukking van type b. Enkele voorbeelden:

- individuele variabelen en individuele constanten zijn van type e;
- alle formules uit de eerste orde predikatenlogica zijn van type t;
- eenplaatsige eerste orde predikaatconstanten en -variabelen zijn van type $\langle e,t\rangle$; als je ze combineert met een individuele variabele of constante als argument—dus met een argument van type e—leveren ze formules op: uitdrukkingen van type t;
- tweeplaatsige eerste orde predikaatconstanten en -variabelen zijn van type $\langle e,\langle e,t\rangle\rangle$; je kunt je voorstellen dat zo'n uitdrukking eerst met een type e uitdrukking combineert tot een eenplaatsig predikaat, dat van type $\langle e,t\rangle$ is (net zoals de transitieve werkwoordsvorm *slaat* combineert met *Jan* als lijdend voorwerp tot de verbale constituent *slaat Jan*, een constituent die dezelfde syntactische status heeft als de intransitieve werkwoordsvorm *slaapt*);
- drieplaatsige eerste orde predikaatconstanten en -variabelen zijn van type $\langle e,\langle e,\langle e,t\rangle\rangle\rangle$ (want met een individuele constante of variabele combineren ze tot een uitdrukking van type $\langle e,\langle e,t\rangle\rangle$, net zoals *geeft aan* combineert met *Jan* tot *geeft aan Jan*, met de status van een transitief werkwoord);
- eenplaatsige predikaatconstanten en -variabelen van de tweede orde hebben type $\langle\langle e,t\rangle,t\rangle$ (want ze combineren met een eenplaatsig eerste orde predikaat, een uitdrukking van type $\langle e,t\rangle$, tot een formule, een uitdrukking van type t);
- het connectief $\neg$ heeft type $\langle t,t\rangle$ (want het combineert met een formule tot een nieuwe formule);
- de connectieven $\vee$, $\wedge$, $\rightarrow$, en $\leftrightarrow$ zouden type $\langle t,\langle t,t\rangle\rangle$ kunnen krijgen (ga na waarom).

De volgende definitie legt de typen vast van uitdrukkingen die worden gevormd met behulp van λ-abstractie.

Definitie 15.7 Lambda abstractie

Als β een uitdrukking is van type b, en v is een variabele van type a, dan is $\lambda v[\beta]$ een uitdrukking van type $\langle a,b\rangle$.

We geven weer een aantal voorbeelden:

- $\lambda x[Lx]$ is van type $\langle e,t\rangle$, want Lx is een formule (heeft dus type t), en x een individuele variabele, die type e heeft. Dit klopt met wat we boven gezien hebben: $\lambda x[Lx]$ is een andere schrijfwijze voor L, een eenplaatsige eerste orde predikaatletter.
- $\lambda Y[Ya]$ is van type $\langle\langle e,t\rangle,t\rangle$, want Y is een predikaatvariabele van type $\langle e,t\rangle$, en Ya is van type t.

– $\lambda\mathbf{X}[\mathbf{X}(P)]$ is van type $\langle\langle\langle e,t\rangle,t\rangle,t\rangle$, want $\mathbf{X}$ is een variabele van type $\langle\langle e,t\rangle,t\rangle$, en $\mathbf{X}(P)$ is een uitdrukking van type t.

In termen van de type-indeling die we zojuist hebben gedefinieerd kunnen we nu ook precies vastleggen wanneer een combinatie van een functor (dat wil zeggen: een naam voor een functie) met een argument welgevormd is. Daartoe moeten de typen van de functor en het argument bij elkaar passen: de functor moet geïnterpreteerd worden als een functie die gedefinieerd is voor de interpretatie van de argument-uitdrukking. Om dat te bewerkstelligen dient de volgende regel voor functor-argument combinatie:

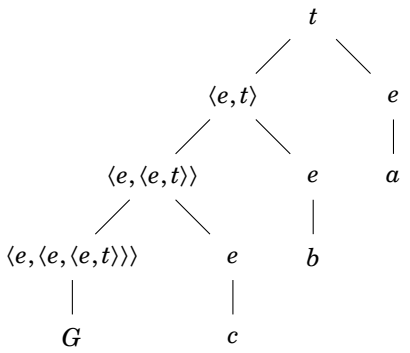
Definitie 15.8 Functor-argument combinatie

Als β een welgevormde typenlogische uitdrukking is van type $\langle a,b\rangle$, en α is een welgevormde typenlogische uitdrukking van type a, dan is $\beta(\alpha)$ een welgevormde typenlogische uitdrukking van type b.

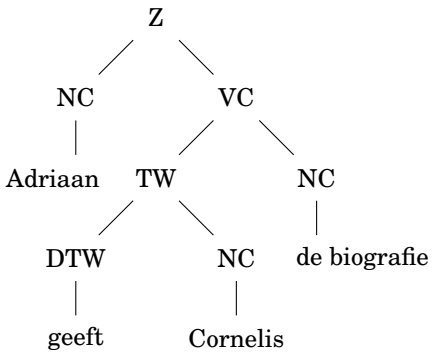
Volgens deze definitie zijn La en Ya, of zo je wilt $L(a)$ en $Y(a)$, welgevormde uitdrukkingen van type t (want: L en Y hebben type $\langle e,t\rangle$, en a heeft type e), net als $\mathbf{X}(P)$ (want: $\mathbf{X}$ heeft type $\langle\langle e,t\rangle,t\rangle$ en P heeft type $\langle e,t\rangle$).

We hebben hierboven gezien dat drieplaatsige eerste orde predikaatletters in de extensionele typenlogica behandeld worden als uitdrukkingen van type $\langle e,\langle e,\langle e,t\rangle\rangle\rangle$. Strikt genomen moeten we nu "*A*driaan geeft de *B*akoenin-biografie aan *C*ornelis" (vergelijk opdrachten 15.23 en 15.24) vertalen als: $G(c)(b)(a)$. Immers, de predikaatletter G combineert eerst met een e-type argument c tot een uitdrukking $G(c)$ van type $\langle\langle e,t\rangle,t\rangle$ (een tweeplaatsige predikaatuitdrukking). De uitdrukking $G(c)$ combineert vervolgens met een e-type argument b tot een uitdrukking $G(c)(b)$ van type $\langle e,t\rangle$. Tenslotte combineert $G(c)(b)$ met een e-type argument a tot een formule (type t uitdrukking) $G(c)(b)(a)$.

In boom-vorm ziet de structuur van $G(c)(b)(a)$ er nu zo uit (de knopen zijn gemarkeerd met de juiste type-aanduidingen):



Vergelijk dit met:



Hier staat TW voor 'transitief werkwoord' (Eng.: TV, transitive verb), en DTW voor 'ditransitief werkwoord' (Eng.: DTV, ditransitive verb). Zoals je ziet gaan we ervan uit dat een ditransitief werkwoord met een meewerkend voorwerp combineert tot een transitief werkwoord. Dit klopt ook, want vervanging van de combinatie *geeft Cornelis* door het transitieve werkwoord *leest* levert weer een syntactisch correcte zin op. Merk op dat de typenlogische boom en de syntactische boom, afgezien van de links-rechts volgorde van de onderwerps-NC en de VC, structureel identiek zijn.

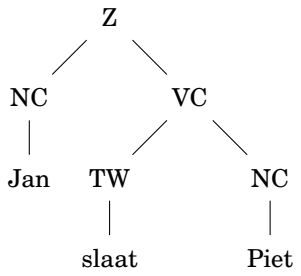
Hoewel uit de notatie $G(c)(b)(a)$ de juiste boomstructuur direct valt af te lezen, is het soms lastig dat de volgorde van de argumenten precies omgekeerd is ten opzichte van de volgorde die we in een eerste orde predikatenlogische formule zouden hebben. De remedie is eenvoudig. We voeren een nieuwe afkorting in. We spreken af dat we desgewenst $G(c)(b)(a)$ mogen schrijven als $G(a,b,c)$ of als $Gabc$. Net zo voor tweeplaatsige predikaten: $R(b)(a)$ kan worden afgekort als Rab.

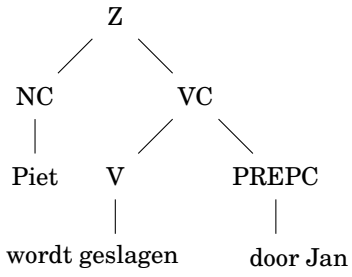
In feite hebben we deze manier van afkorten hierboven al gebruikt: atomaire predikatenlogische formules werden zonder haakjes in de typenlogica opgenomen. Ook de vertaling van "Opstaan voor iemand misstaat niemand" die we in § 15.4 hebben gegeven, $\neg\exists z\mathbf{M}(\lambda x[\exists yOxy],z)$, is strikt genomen een afkorting voor: $\neg\exists z\mathbf{M}(z)(\lambda x[\exists yO(y)(x)])$. Merk op: $\lambda x[\exists yOxy]$ is de vertaling van het *onderwerp* van de zin "Opstaan voor iemand misstaat niemand". De weggelaten haakjes kwamen verder, met name bij herhaalde λ-conversie, weer terug om duidelijk te maken wat de nieuwe functie-applicaties waren.

Opgave 15.26 Geef de typen van de constanten die we in § 15.4 hebben gebruikt voor de vertaling van *verstandig zijn*, *strafbaar zijn*, *misstaan*, en *plicht zijn*.

Opgave 15.27 Geef het type van de vertaling van *snel* in "Jan loopt snel" (zie de bespreking op bladzijde 179 en volgende).

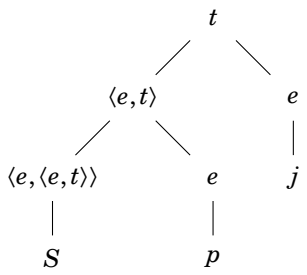
We kunnen nu ook laten zien hoe λ-abstractie gebruikt kan worden om argument-plaatsen te verwisselen. In feite is het onderscheid tussen de actieve en passieve vorm van een Nederlandse zin niets anders dan het omdraaien van twee argumenten. Vergelijk de structuurbomen voor "Jan slaat Piet" en "Piet wordt geslagen door Jan":





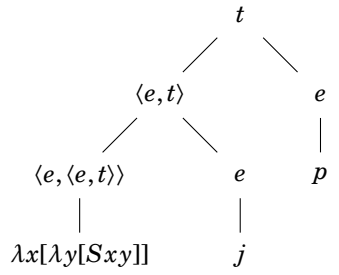
Hier staat V voor *verbum* en PREPC voor *prepositie-constituent* (Eng.: PP, prepositional phrase).

Neem aan dat het transitieve werkwoord *slaan* in de typenlogica een vertaling S heeft, van type $\langle e,\langle e,t\rangle\rangle$. De vertaling van "Jan slaat Piet" wordt dan: $S(p)(j)$, met de volgende structuurboom:



Stel nu dat we uit $S(p)(j)$—of, zo je wilt, afgekort: Sjp—door middel van dubbele λ-abstractie een nieuwe predikaatuitdrukking van type $\langle e,\langle e,t\rangle\rangle$ zouden vormen, als volgt. Eerst passen we λ-abstractie toe op het lijdend voorwerp. Dit geeft: $\lambda y[S(y)(j)]$, een uitdrukking van type $\langle e,t\rangle$. We kunnen dit uiteraard ook afgekort opschrijven: $\lambda y[Sjy]$. De uitdrukking betekent *door Jan geslagen worden*. Vervolgens passen we λ-abstractie toe op het onderwerp van de oorspronkelijke zin. Dit geeft $\lambda x[\lambda y[S(y)(x)]]$, afgekort $\lambda x[\lambda y[Sxy]]$. Dit is een uitdrukking van type $\langle e,\langle e,t\rangle\rangle$ (ga dit na), met als betekenis *geslagen worden*.

Met behulp van de uitdrukking die we zojuist hebben gefabriceerd kunnen we nu de passieve zin "Piet wordt geslagen door Jan" vertalen. De vertaling wordt: $\lambda x[\lambda y[Sxy]](j)(p)$. Deze vertaling heeft een structuurboom die correspondeert met de syntactische structuurboom voor "Piet wordt geslagen door Jan":



De typenlogische verantwoording van het feit dat de twee zinnen "Jan slaat Piet" en "Piet wordt geslagen door Jan" hetzelfde betekenen is nu simpel. De vertaling van "Piet wordt geslagen door Jan" kan als volgt worden vereenvoudigd met behulp van λ-conversie: $\lambda x[\lambda y[Sxy]](j)(p)$ wordt geconverteerd tot $\lambda y[Sjy](p)$, en dit wordt weer geconverteerd tot Sjp. De vertaling is dus hetzelfde als die van "Jan slaat Piet".

Opgave 15.28 Zij gegeven dat "*A*driaan geeft de *B*akoenin-biografie aan *C*ornelis" de vertaling $G(c)(b)(a)$ heeft, of afgekort: $Gabc$. Vertaal nu in uitdrukkingen van de extensionele typenlogica, en geef van elke typenlogische uitdrukking het type aan:

1. de Bakoenin-biografie geven aan Cornelis
2. geven aan Cornelis
3. de Bakoenin-biografie krijgen van Cornelis
4. krijgen van Cornelis
5. krijgen van iemand
6. aan Cornelis iets geven
7. van iedereen iets krijgen.

Opgave 15.29 Het gegeven is als in de vorige opdracht. Geef van elk van de volgende uitdrukkingen het type aan, en geef een parafrase in het Nederlands:

1. $\lambda x[\lambda y[Gybx]]$
2. $\lambda x[\lambda y[Gxby]]$
3. $\lambda x[\lambda y[\exists z Gyzx]]$
4. $\lambda x[\lambda y[\exists z Gxzy]]$.

Wie een taalkundige toepassing wil zien van deze λ-trucs leze [Dowty 1982].

Het wordt hoog tijd dat we een enkel woord wijden aan de *semantiek* van de extensionele typenlogica. Die semantiek is ingewikkelder dan die van de eerste orde predikatenlogica, omdat we voor elk type a een domein D_a nodig hebben van objecten van type a. Dit lijkt erger dan het is. We hoeven al die domeinen niet expliciet in te voeren, maar we kunnen ze recursief definiëren, uitgaande van de domeinen voor de basistypen e en t. Het domein D_t voor het basistype t is geen probleem: dat bestaat gewoon uit de twee waarheidswaarden ONWAAR en WAAR, en we kunnen er de verzameling $\{0,1\}$ voor

gebruiken. Het enige domein dat expliciet gegeven moet zijn is het domein D_e van de individuele objecten.

We frissen even ons geheugen op wat betreft de verzamelingstheoretische notatie. Als X en Y verzamelingen zijn, dan gebruiken we X^Y voor de verzameling van alle functies f met domein Y en co-domein X. De notatie $f : Y \to X$ stond voor: f is een functie met domein Y en co-domein X. Dat X het co-domein is van f wil niet zeggen dat elk element van X ook als beeld onder f hoeft op te treden.

De recursieve definitie van de domeinen van de verschillende mogelijke typen gaat nu als volgt:

Definitie 15.9 Domeinen van de typen *in een typenlogische taal met basistypen e en t.*

- $D_t = \{0, 1\}$*;*
- D_e *moet expliciet worden gegeven;*
- *Als de domeinen van de typen a en b respectievelijk D_a en D_b zijn, dan is het domein voor het type $\langle a, b\rangle$ de volgende verzameling:*

$$D_b^{D_a}$$

Met andere woorden: de objecten van type $\langle a, b\rangle$ zijn de functies van D_a naar D_b.

- *Niets anders is een domein voor een type.*

We kunnen nu een *model* voor een extensionele typenlogische taal definiëren als een paar $\langle D_e, I\rangle$, waarbij D_e een verzameling individuen is, en I een interpretatie-functie die, voor elk type a, aan alle constanten van type a een element toekent uit het domein D_a dat geconstrueerd is zoals hierboven is aangegeven.

Als je vind dat dit allemaal nogal abstract klinkt, dan sta je waarschijnlijk niet alleen, en het wordt je ook niet kwalijk genomen. Verder is het—zoals zoveel dingen in het leven—vooral een kwestie van wennen. Om dat gewenningsproces te stimuleren kijken we nog even naar de domeinen van een paar simpele typen.

Om de gedachten te bepalen: neem aan dat D_e slechts drie elementen heeft: $\{\star, \circ, \bullet\}$. Verder weten we: $D_t = \{0, 1\}$. Hoe ziet $D_{\langle e,t\rangle}$ (dat wil zeggen: het domein van de objecten van type $\langle e, t\rangle$, de eenplaatsige eerste orde predikaten) er nu uit? Volgens de definitie van de domeinen hebben we:

$$D_{\langle e,t\rangle} = D_t^{D_e} = \{0, 1\}^{\{\star,\circ,\bullet\}}$$

Dus: $D_{\langle e,t\rangle}$ is de verzameling *karakteristieke functies* van de verzameling $\{\star, \circ, \bullet\}$. Elk van die functies karakteriseert een deelverzameling van D_e, dus we kunnen zeggen dat $D_{\langle e,t\rangle}$ neerkomt op de verzameling van alle deelverzamelingen van het domein D_e. Een interpretatiefunctie I voor een typenlogische taal kent aan elke constante van type $\langle e, t\rangle$ een element van $D_{\langle e,t\rangle}$ toe. Eenplaatsige eerste orde predikaatconstanten hebben type $\langle e, t\rangle$. Laat L zo'n predikaatconstante zijn, en veronderstel dat de interpretatiefunctie I aan L de volgende karakteristieke functie toekent:

$$\begin{array}{ccc} \star & \longrightarrow & 0 \\ \circ & \longrightarrow & 1 \\ \bullet & \longrightarrow & 1 \end{array}$$

Deze karakteristieke functie is een element van $D_{\langle e,t\rangle}$, en ze karakteriseert de deelverzameling $\{\circ, \bullet\}$ van D_e. Vergelijk dit met de interpretatiefunctie van een eerste orde model, dat aan eenplaatsige predikaatletters deelverzamelingen van het individuendomein toekent: hier zou L meteen worden geïnterpreteerd als de deelverzameling $\{\circ, \bullet\}$ van het domein. Beide manieren van interpreteren komen natuurlijk op hetzelfde neer.

Nog een voorbeeld. Hoe ziet $D_{\langle e,\langle e,t\rangle\rangle}$ eruit? We gaan weer te werk volgens het boekje (dat wil zeggen: volgens de definitie van de typen-domeinen):

$$D_{\langle e,\langle e,t\rangle\rangle} = (D_{\langle e,t\rangle})^{D_e} = (\{0, 1\}^{\{\star,\circ,\bullet\}})^{\{\star,\circ,\bullet\}}$$

De interpretatiefunctie I zou nu aan een constante R van type $\langle e, \langle e, t\rangle\rangle$ de volgende interpretatie kunnen toekennen:

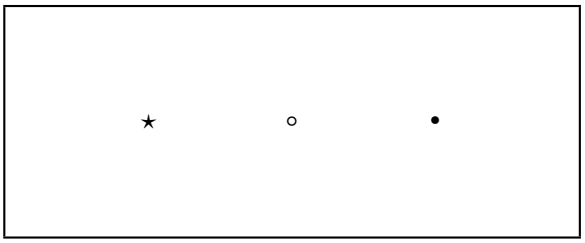
$$\star \longrightarrow \begin{pmatrix} \star & \longrightarrow & 0 \\ \circ & \longrightarrow & 1 \\ \bullet & \longrightarrow & 1 \end{pmatrix}$$

$$\circ \longrightarrow \begin{pmatrix} \star & \longrightarrow & 0 \\ \circ & \longrightarrow & 1 \\ \bullet & \longrightarrow & 0 \end{pmatrix}$$

$$\bullet \longrightarrow \begin{pmatrix} \star & \longrightarrow & 1 \\ \circ & \longrightarrow & 1 \\ \bullet & \longrightarrow & 1 \end{pmatrix}$$

Ga zelf na dat dit inderdaad een plaatje is van een element van $D_{\langle e,\langle e,t\rangle\rangle}$.

Opgave 15.30 In een eerste orde model zou R gewoon worden geïnterpreteerd als een tweeplaatsige relatie op $\{\star, \circ, \bullet\}$. Teken die relatie met behulp van pijlen in de volgende figuur:



Als laatste voorbeeld beschouwen we het domein $D_{\langle\langle e,t\rangle,t\rangle}$. Weer passen we de definitie toe:

$$D_{\langle\langle e,t\rangle,t\rangle} = D_t^{D_{\langle e,t\rangle}} = D_t^{(D_t^{D_e})} = \{0, 1\}^{(\{0,1\}^{\{\star,\circ,\bullet\}})}$$

Tekenen van een element van dit domein wordt ons teveel werk, maar we willen wel even met je over zo'n element *praten*. Neem aan dat s een constante is van type e zodanig dat $I(s) = \star$. Dan is $\lambda Y[Y(s)]$ een uitdrukking die moet worden geïnterpreteerd als de karakteristieke functie in $D_{\langle\langle e,t\rangle,t\rangle}$ die elke karakteristieke functie in $D_{\langle e,t\rangle}$ die $\star$ afbeeldt op 1, afbeeldt op 1, en elke andere karakteristieke functie in $D_{\langle e,t\rangle}$ op 0. Moet je hier even op

mediteren? Ga gerust uw gang: de meeste stervelingen hebben wel wat tijd nodig voordat ze de Typentheoretische Verlichting bereiken. Anders gezegd: $\lambda Y[Y(s)]$ wordt geïnterpreteerd als (de karakteristieke functie van) de verzameling van alle eerste orde eigenschappen van $\star$. Een eerste orde eigenschap van $\star$ is niets anders dan (de karakteristieke functie van) een deelverzameling van $\{\star, \circ, \bullet\}$ waar $\star$ element van is.

De bedoeling van het bovenstaande verhaal was: je een indruk geven van de semantiek van de extensionele typenlogica. Het verhaal kan (en moet) uiteraard veel systematischer en uitvoeriger worden verteld. Zie voor meer informatie: [Gamut 1982], deel 2. Tenslotte dient nog vermeld dat de λ-calculus ook ten grondslag ligt aan de programmeertaal LISP, die vooral populair is bij onderzoekers op het terrein van de *kunstmatige intelligentie* (Eng.: *artificial intelligence*), waar geprobeerd wordt computers 'intelligent gedrag' bij te brengen. Voor informatie over LISP zij verwezen naar [Steels 1983].

15.6 Modale predikatenlogica

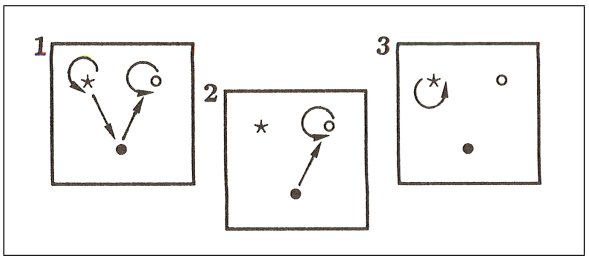
Modale predikatenlogica (gangbare afkorting: MPL) ontstaat door $\Box$ en $\Diamond$ toe te voegen aan de verzameling operatoren van de gewone eerste orde predikatenlogica. Ze verhoudt zich dus tot de eerste orde predikatenlogica zoals de modale propositielogica (de gangbare afkorting daarvoor is MpL) zich verhoudt tot de gewone propositielogica. $\Box$ en $\Diamond$ zijn eenplaatsige operatoren op formules, net als het negatieteken.

Opgave 15.31 Geef een recursieve definitie van de verzameling welgevormde formules van de modale predikatenlogica.

$\Box\varphi$ betekent: φ is noodzakelijk waar. We leggen dit weer uit als: φ is waar *in alle mogelijke werelden*. $\Diamond\varphi$ staat voor: φ is mogelijkerwijs waar, of: het is mogelijk dat φ. De uitleg: φ is waar *in minstens één mogelijke wereld*.

Een *model* voor modale predikatenlogica bestaat nu uit een verzameling *contexten*, *indices*, of *mogelijke werelden* (in dit verband allemaal namen voor hetzelfde), laten we zeggen W, zo dat met elk element $w \in W$ een predikatenlogisch model oude-stijl is geassocieerd. Wanneer we aannemen dat elke wereld hetzelfde individuendomein heeft kunnen we zo'n model definiëren als een drietal $\langle D, W, I\rangle$, waarbij D een individuendomein is, W een verzameling mogelijke werelden, en I een interpretatiefunctie die nu iets ingewikkelder is dan in gewone eerste orde predikatenlogische modellen. I kent aan alle predikaatletters *per wereld* een interpretatie toe. Dus: $I_w(P)$ hoeft niet hetzelfde te zijn als $I_{w'}(P)$ (aangenomen dat w en w' twee verschillende elementen van W zijn).

Aanschouwelijk onderricht werkt weer het beste, dus een plaatje van een zeer eenvoudig model voor een modale predikatenlogische taal is hier op zijn plaats. Het model dat hier is afgebeeld geeft een beeld van de situatie rond drie objecten in drie verschillende mogelijke werelden:



Neem aan dat de getekende pijltjes de interpretatie vormen voor een tweeplaatsige predikaatletter R. Merk op dat de interpretatie *per wereld* kan worden omschreven als een verzameling geordende paren uit het domein (een tweeplaatsige relatie op $D = \{\star, \circ, \bullet\}$).

We kunnen nu formules uit een modale predikatenlogische taal met R als predikaatletter gaan *evalueren* (op waarheid of onwaarheid onderzoeken) in een mogelijke wereld in het hier getekende model. Bij voorbeeld: is $\exists xRxx$ waar in wereld 3 van het model? Antwoord: ja, want $\star$ staat in die wereld in de pijlrelatie tot zichzelf. Is $\exists xRxx$ waar in wereld 1? Ja, kijk maar. Nu met modale operatoren. Is $\Box\exists xRxx$ waar in wereld 1? Ja, want in alle mogelijke werelden (dat wil in dit model zeggen: in wereld 1, wereld 2 en wereld 3) is $\exists xRxx$ waar. Is $\exists x\Box Rxx$ waar in wereld 1? Nee, want er is geen enkel object dat zowel in wereld 1, in wereld 2 en in wereld 3 in de pijlrelatie tot zichzelf staat.

Het model laat zien dat $\Box\exists xRxx$ en $\exists x\Box Rxx$ *niet* equivalent zijn. Intuïtief klopt dat ook wel, want vergelijk het volgende huis, tuin en keuken voorbeeld: "Iemand moet dit spelletje gaan winnen". Dit kan twee dingen betekenen, en elk van die lezingen vraagt om een andere vertaling in de modale predikatenlogica. De twee formules zijn: $\exists x\Box Sx$ en $\Box\exists xSx$ (met S voor 'gaat dit *S*pelletje winnen').

Opgave 15.32 Geef een ondubbelzinnige parafrase in het Nederlands van elk van deze twee mogelijke lezingen.

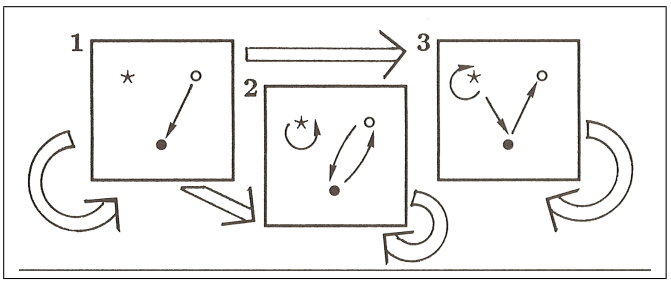
Opgave 15.33 Welke twee lezingen heeft "Iedereen kan dit spelletje verliezen"? Geef voor elk van een beide lezingen een vertaling in de modale predikatenlogica.

Opgave 15.34 Kijk weer naar het hierboven getekende model voor een modale predikatenlogische taal met tweeplaatsige predikaatletter R. Welke van de volgende formules zijn waar in wereld 1 van dit model:

1. $\Box\forall x\exists yRxy$
2. $\Box\exists x\forall yRxy$
3. $\exists x\Box\forall yRxy$
4. $\forall x\Box\exists yRxy$.

De semantiek voor modale logica wordt interessanter wanneer je ook nog *structuur* aanbrengt in de verhoudingen tussen de verschillende mogelijke werelden. Neem aan dat werelden al dan niet *toegankelijk* kunnen zijn vanuit bepaalde andere werelden. Daartoe voeren we

een tweeplaatsige relatie T in op de verzameling werelden W. Een model voor modale predikatenlogica krijgt nu de vorm $\langle D, W, T, I\rangle$. Een plaatje van zo'n model:



De pijl $\Longrightarrow$ staat voor de toegankelijkheidsrelatie T. Zoals je ziet is elke wereld in dit model toegankelijk vanuit zichzelf. Met andere woorden: in dit model is de toegankelijkheidsrelatie *reflexief*. Verder is wereld 2 toegankelijk vanuit wereld 1, maar niet andersom, en wereld 3 is toegankelijk vanuit 1. De waarheidsclausule voor $\Box$ en $\Diamond$ wordt nu:

- $\Box\varphi$ is waar in wereld w desda φ waar is in alle werelden die toegankelijk zijn vanuit w (dus: alle werelden waar vanuit w een pijl $\Longrightarrow$ naar wijst).
- $\Diamond\varphi$ is waar in wereld w desda φ waar is in minstens één wereld die toegankelijk is vanuit w.

Opgave 15.35 Beschouw het model dat hierboven gegeven is.

1. Is $\Box\exists xRxx$ waar in wereld 1 van dit model?
2. En in wereld 2?
3. Is $\Box\Box\exists xRxx$ waar in wereld 3?

Het aardige aan het invoeren van de toegankelijkheidsrelatie T is nu dat we de betekenis van 'noodzakelijk' en 'mogelijk' nader kunnen specificeren door bepaalde *eisen* op te leggen aan de toegankelijkheidsrelatie. Als we bij voorbeeld afspreken dat in alle modellen die we willen beschouwen de toegankelijkheidsrelatie T *reflexief* moet zijn (dat wil zeggen dat in al die modellen geldt dat alle mogelijke werelden er toegankelijk zijn vanuit zichzelf), dan betekent dat elke formule van de volgende vorm universeel geldig wordt (dat wil zeggen: waar in elke wereld van elk model):

$$\varphi \rightarrow \Diamond\varphi.$$

Merk op dat dit niet een formule is maar een formule-schema: afhankelijk van wat we voor φ invullen krijgen we telkens een andere formule. Dat het schema geldig is valt als volgt in te zien. Voor elk model en voor elke mogelijke wereld geldt: als φ waar is in w, dan is er een mogelijke wereld die toegankelijk is vanuit w waarin φ waar is, namelijk w zelf, dankzij de reflexiviteit van de toegankelijkheidsrelatie in het model. Dus $\Diamond\varphi$ is ook waar in w.

Opgave 15.36 Laat zien: als we de toegankelijkheidsrelatie T *symmetrisch* maken, dan geldt het schema $\varphi \rightarrow \Box\Diamond\varphi$.

Opgave 15.37 Welk schema geldt er als T *transitief* is?

Over mogelijke eisen op de toegankelijkheidsrelatie valt nog veel meer interessants te zeggen. Wie meer wil weten over de semantiek voor MPL zij verwezen naar [Hughes & Cresswell 1968], en naar [Gamut 1982], deel 2. Beide boeken geven ook allerlei (taal)filosofische achtergronden.

Behalve de semantische kijk op modale propositie- en predikaten-logica is er weer de axiomatische: een gangbaar axioma voor MpL en MPL is

$$\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi).$$

Naast Modus Ponens hebben axiomatische systemen voor MpL en MPL meestal een afleidingsregel als "wanneer φ een axioma is, dan $\Box\varphi$ ook".

Door $\Box$ en $\Diamond$ op een specifieke manier te interpreteren ontstaan allerlei toepassingen van MpL en MPL. Een voorbeeld daarvan, de *tijdslogica* die ontstaat door eenplaatsige tijdsoperatoren te introduceren hebben we in § 9.8 al aangestipt. De modellen maken gebruik van een *eerder dan* relatie tussen tijdstippen. De operator F, voor '*eens* in de toekomst ...' is qua betekenis analoog aan $\Diamond$. $\Box$ heeft ook een analogon, namelijk de operator G, voor '*altijd* in de toekomst ...' (G is een afkorting voor 'it is *g*oing to be the case'). Net zo hebben we een 'universele' en een 'existentiële' tijdsoperator voor verleden tijd: P staat voor '*eens* in het verleden ...'; H voor '*altijd* in het verleden ...' (H is een afkorting voor 'it *h*as been the case'). Verder kunnen modaal-logische systemen bij voorbeeld worden gebruikt om de logica van *weten* en *geloven* bloot te leggen.

Filosofisch is er veel gekrakeel geweest rond de implicaties van de mogelijke werelden metafoor als uitleg voor 'noodzakelijkheid'. Wanneer je weet dat 'noodzakelijkheid' een kernbegrip is in allerlei vormen van metafysica, dan kun je je voorstellen dat discussies over wat het *betekent* dat iets 'anders had kunnen zijn dan het in feite is' kunnen voeren tot grote hoogten van filosofische spitsvondigheid. Betekent 'stand van zaken A had anders kunnen zijn' bij voorbeeld hetzelfde als '*A* is *a posteriori* waar'? En is dat weer hetzelfde als of iets anders dan '*A* is een *synthetische bewering*'?

15.7 Intensionele typenlogica

Intensionele logica is een andere benaming voor modale logica (of eigenlijk: voor vormen van logica waarin de semantiek gebruik maakt van mogelijke werelden). *Intensionele typenlogica* ontstaat door het combineren van typenlogica en modale logica. Het gebruik van deze combinatie voor het weergeven van de betekenis van uitdrukkingen uit de natuurlijke taal is voorgesteld door de logicus Richard Montague. Zie [Montague 1974], maar

veel toegankelijker is [Gamut 1982], deel 2. Montague's voorstellen hebben geleid tot een aparte school in het onderzoek van natuurlijke taal, de zogenaamde Montague-grammatica.

Montague maakt de stap van extensionele naar intensionele typenlogica omdat hij de zogenaamde 'intensionele verschijnselen' in de natuurlijke taal wil kunnen vangen in de logische vertalingen voor uitdrukkingen uit de natuurlijke taal. Die logische vertalingen worden dan vervolgens geïnterpreteerd in modellen voor intensionele typenlogica.

Wellicht het beroemdste voorbeeld van een intensioneel verschijnsel is Frege's *ochtendster-avondster paradox*. In de Oudheid hadden de ochtendster (dat wil zeggen het laatst verdwijnende lichtpuntje in het Oosten aan de ochtendhemel) en de avondster (dat wil zeggen het eerst verschijnende lichtpuntje in het Westen aan de avondhemel) verschillende namen. Toen ontdekten de Babyloniërs dat het hier om een en hetzelfde hemellichaam gaat (namelijk de planeet Venus). Dus:

De Babyloniërs ontdekten dat de ochtendster de avondster is.

Dat klopt ook:

De ochtendster is de avondster.

Maar als de twee termen *de ochtendster* en *de avondster* niets anders zijn dan verwijzingen naar een en hetzelfde object, dan zou je ze met behoud van waarheidswaarde voor elkaar moeten kunnen substitueren. En dan zou je moeten kunnen zeggen:

De Babylonië

Maar deze zin is *niet* waar, want dat de avondster identiek is aan zichzelf wisten de Babyloniërs natuurlijk altijd al. De context *De Babyloniërs ontdekten dat...* is blijkbaar een omgeving waarbinnen het principe van substitutie met behoud van waarheidswaarde voor termen die naar hetzelfde object verwijzen *niet* meer opgaat. Zo'n context noemen we een *intensionele context* (Eng.: intensional context), of ook wel: een *referentieel ondoorzichtige context* (Eng.: referentially opaque context).

Intensioneel is de tegenhanger van *extensioneel*: een context waarbinnen het substitutie-principe als boven wel opgaat heet *extensioneel*. Een andere benaming is: *referentieel doorzichtige context* (Eng.: referentially transparent context). Ook de contexten *Het is noodzakelijk dat...* en *Het is mogelijk dat...* zijn intensioneel. Immers, uit:

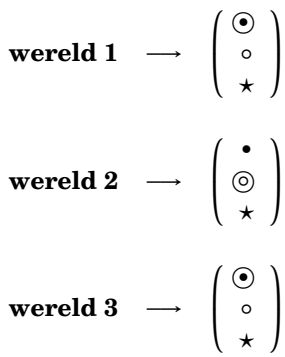
Het is noodzakelijk dat de logicadocenten de stof goed uitleggen.

volgt nog niet:

Het is noodzakelijk dat Kleene en Tarski de stof goed uitleggen.

ook al is het in de mogelijke wereld waarin wij ons bevinden (de werkelijke wereld) zo dat Kleene en Tarski de logica-docenten zijn. Immers, in andere mogelijke werelden (in andere voorstelbare standen van zaken) hoeft het niet zo te zijn dat Kleene en Tarski de logica-docenten zijn. In *die* mogelijke werelden moeten niet Kleene en Tarski de stof goed uitleggen, maar degenen die *daar* de logica-docenten zijn.

De bovenstaande parafrase geeft al aan in welke richting we de oplossing moeten zoeken. Blijkbaar moeten we met *de avondster*, *de ochtendster*, *de logica-docent* niet zonder meer een individu associëren, maar veeleer een *functie* die in elke mogelijke wereld een individu oplevert. Dus: *binnen* een mogelijke wereld is de verwijzing van deze termen een individu (bij voorbeeld: de planeet Venus, of Jan), maar *globaal* is het een functie van de verzameling W van mogelijke werelden naar individuen in die werelden. De verwijzing-in-engere-zin noemen we de *extensie* van de term. De 'globale' verwijzing noemen we de *intensie*. Frege's ochtendster-avondster paradox kan nu worden opgelost door over te stappen van extensies naar intensies. Weer is een plaatje duizend woorden waard. Een plaatje voor de *intensie* van *de logica-docent*:



Zoals je ziet: een functie van de verzameling mogelijke werelden W = {wereld 1, wereld 2, wereld 3} naar individuen in die werelden. De *extensie* van *de logica-docent* is in wereld 1 de waarde die de *intensie*-functie daar oplevert; in wereld 2 en in wereld 3 evenzo.

Het intensie-extensie onderscheid kunnen we ook goed gebruiken voor de analyse van zinnen als:

Marietje zoekt de logica-docent.

Het zou kunnen wezen dat Marietje helemaal niet weet dat Jan de logica-docent is. Ze loopt gewoon te zoeken naar de persoon die het vak logica verzorgt, wie dat ook moge zijn. Hieraan zien we dat we uit bovenstaande zin en

Jan is de logica-docent.

niet mogen concluderen tot:

Marietje zoekt Jan.

Conclusie: ook de context *Marietje zoekt...* is intensioneel. De technische oplossing die door Montague is voorgesteld maakt weer gebruik van mogelijke werelden. Neem aan dat *zoeken* wordt geïnterpreteerd als een relatie tussen individuen (de zoekers) en *intensies van individuen* (die

dan staan voor ‘concepties’ van wat wordt gezocht). Slag om de arm: hier komt het globaal op neer; Montague maakt het allemaal nog veel ingewikkelder, maar wij beperken ons hier tot de Hoofdgedachten van de Meester.

Dat het onderscheid tussen een *intensionele* en een *extensionele* lezing van *zoeken* taalkundig van belang is moge blijken uit het feit dat sommige talen er twee verschillende constructies voor hebben. Vergelijk:

Jean cherche une amie qui est *blonde.*

Jean cherche une amie qui soit *blonde.*

Het zoeken in de eerste zin is extensioneel: er wordt een individu gezocht. In de tweede voorbeeldzin is het zoeken intensioneel: er wordt een ‘conceptie’ gezocht, en misschien—wie zal het zeggen—een hersenschim nagejaagd. In het Nederlands hebben we geen *subjonctif*, maar we kunnen het verschil met wat kunst-en vliegwerk ook wel uitdrukken in de vertalingen: “Jan zoekt een vriendin, en ze is blond” versus “Jan zoekt een vriendin die blond moet zijn”. Op de technische details van de semantiek voor intensionele typenlogica gaan we hier niet in. Het komt neer op combineren van de ingrediënten voor de semantiek van modale logica met die voor de semantiek voor extensionele typenlogica, iets wat niet al te moeilijk is voor een kok die al weet hoe hij deze twee gerechten afzonderlijk moet toebereiden.

Deel IV

Toepassingen

Hoofdstuk 16

Logisch programmeren

Een van de meest praktische toepassingen van logica in de informatica is het gebruik van de logica zelf als programmeertaal. Heel in het algemeen gaat programmeren met logica zo. We nemen een eindige *consistente* verzameling formules in een geschikte taal; dit is ons programma Π. We voegen daar een nieuwe formule $\neg\varphi$ aan toe en we laten het systeem uitmaken of $\Pi \cup \{\neg\varphi\}$ nog steeds consistent is. Als dat niet zo is dan is φ kennelijk waar in alle modellen van Π, ofwel: φ volgt logisch uit Π. Als het wel zo is dan volgt φ *niet* uit Π.

Om één en ander te laten werken moeten de formules uit Π een vorm hebben die garandeert dat eindige consistentie voor dergelijke formules beslisbaar is. De poging om verzameling Π inconsistent te maken door toevoegen van een nieuwe formule heeft immers alleen zin wanneer we weten dat Π zelf consistent is. Verder moeten de formules $\neg\varphi$ die we toevoegen een vorm hebben die garandeert dat de vraag of ze consistent zijn met eindige verzamelingen programma-formules beslisbaar is.

Propositielogica is te arm om als programmeertaal te worden gebruikt; predikatenlogica is daarentegen juist te rijk, want een minimum-eis is dat onze programma's consistent zijn, en de vraag of een eindige verzameling predikatenlogische zinnen consistent is, is in het algemeen onbeslisbaar. Het is dus zaak om een deel van de predikatenlogica af te zonderen met de gewenste eigenschappen.

PROLOG staat voor programmeren in logica. De ingrediënten van PROLOG programma's zijn predikatenlogische zinnen van een speciale vorm.

Definitie 16.1 *Een* **Horn zin** *is een formule van de vorm*

$$\forall x_1 \cdots \forall x_m((A_1 \wedge \cdots \wedge A_n) \rightarrow A)$$

of van de vorm

$$\forall x_1 \cdots \forall x_m(\neg A_1 \vee \cdots \vee \neg A_n),$$

waarbij $A, A_1, \ldots, A_n$ atomaire predikatenlogische formules zijn waarbinnen hoogstens de variabelen $x_1, \ldots, x_m$ vrij voorkomen.

De eigenschappen van Horn zinnen werden bestudeerd (door de logicus A. Horn) lang voor de uitvinding van PROLOG. In PROLOG is een speciale notatie voor Horn zinnen in zwang.

$$\forall x_1 \cdots \forall x_m((A_1 \wedge \cdots \wedge A_n) \rightarrow A)$$

wordt genoteerd als

$$A \text{ :- } A_1, \ldots, A_n. \tag{1}$$

Een Horn zin van de vorm (1) wordt een *programma-clausule* (Eng.: clause) genoemd. De logische formule

$$\forall x_1 \cdots \forall x_m(\neg A_1 \vee \cdots \vee \neg A_n)$$

wordt genoteerd als

$$\text{:- } A_1, \ldots, A_n. \tag{2}$$

Een Horn zin van de vorm (2) heet een *doelclausule*. De onderdelen A_1 tot en met A_n van een doelclausule heten de *subdoelen* van de doelclausule.

Als we een Horn zin in het algemeen opvatten als een implicatie, dan kan een doelclausule worden gezien als een implicatie met een lege consequent:

$$\forall x_1 \cdots \forall x_m(\neg A_1 \vee \cdots \vee \neg A_n) \tag{3}$$

is immers equivalent met

$$\forall x_1 \cdots \forall x_m(\neg A_1 \vee \cdots \vee \neg A_n \vee \bot),$$

wat weer kan worden geschreven als:

$$\forall x_1 \cdots \forall x_m((A_1 \wedge \cdots \wedge A_n) \rightarrow \bot).$$

Hierbij staat $\bot$ voor een contradictie; een altijd onwaar disjunct kan immers zonder bezwaar worden weggelaten. Een andere schrijfwijze voor (3) is:

$$\neg\exists x_1 \cdots \exists x_m(A_1 \wedge \cdots \wedge A_n).$$

Aan deze notatie kunnen we zien dat doelclausules de vorm hebben van *negaties*. Wanneer $n = 0$ in een doelclausule dan noteren we dit als

:-

of

$\Box$.

Deze formule heet de *lege clausule* (Eng.: empty clause), en zij staat eveneens voor een contradictie: het is de PROLOG notatie voor $\bot$. De consequent en de antecedent van de lege clausule zijn beide leeg, hetgeen betekent dat er niets volgt uit de lege antecedent; dit is altijd onwaar.

Wanneer $n = 0$ in een programma-clausule hebben we het volgende

A :-.

Dit wordt ook wel genoteerd als:

A.

Een programma-clausule van deze vorm heet een *programma-feit*. Een feit is een programma-clausule met een lege antecedent. Terugvertaald naar standaardnotatie ziet (16) er zo uit:

$$\forall x_1 \cdots \forall x_m A.$$

Een programma-clausule van de vorm (1) met $n \neq 0$ (een programma-clausule met een niet-lege antecedent) heet een *programma-regel*. In de programma-regel (1) vormt A de *kop* en $A_1, \ldots, A_n$ de *staart*. Een feit is een programma-clausule met een lege staart.

We kunnen nu een definitie geven van wat we verstaan onder een PROLOG programma.

Definitie 16.2 *Een PROLOG programma is een eindige verzameling van programma-clausules.*

Voorbeeld 16.1 Het volgende PROLOG programma bevat informatie over ons koninklijk huis (in standaard PROLOG notatie beginnen individuele variabelen met hoofdletters of het onderstrepings-teken, en individuele constanten met kleine letters):

```
man(willem_alexander).
man(claus).
man(bernhard).

vrouw(beatrix).
vrouw(juliana).

ouder_van(claus,willem_alexander).
ouder_van(beatrix,willem_alexander).
ouder_van(juliana,beatrix).
ouder_van(bernhard,beatrix).

vader_van(X,Y) :-
   man(X), ouder_van(X,Y).

moeder_van(X,Y) :-
   vrouw(X), ouder_van(X,Y).
```

De eerste negen programma-clausules in dit voorbeeld zijn feiten, de twee overige zijn regels. Een eindige verzameling van PROLOG feiten is niets anders dan een zogenaamd *relationeel gegevensbestand*; een efficiënt PROLOG systeem is geknipt voor het manipuleren van relationele gegevensbestanden.

Wanneer het programma uit voorbeeld 16.1 is geladen kunnen we vragen stellen met behulp van doelclausules. Een voorbeeld van een ja–nee vraag:

```
| ?- man(willem_alexander).

yes
| ?-
```

Hier is `| ?-` de PROLOG prompt. PROLOG interpreteert alles wat de gebruiker intikt als doelclausule.

Vraagwoord-vragen kunnen worden gesteld met behulp van doelclausules die variabelen bevatten. Een mogelijke vraag is bij voorbeeld: *Wie zijn mannen?*

```
| ?- man(X).

X = willem_alexander
```

Het systeem wacht na het geven van de eerste passende `X`; de gebruiker kan opdracht geven tot verder zoeken door intikken van `;`. Dit levert:

```
| ?- man(X).

X = willem_alexander ;

X = claus ;

X = bernhard ;

no
| ?-
```

Na het derde verzoek om verder speurwerk geeft het systeem te kennen dat er geen nieuwe antwoord-substituties te vinden zijn.

Merk op dat de regels in Voorbeeld 16.1 enkel variabelen bevatten. In een regel kunnen ook individuele constanten voorkomen. Bij voorbeeld:

```
kind_van_bernhard(X) :-
   ouder_van(bernhard,X).
```

Regels van de meer algemene vorm zijn echter nuttiger. De vraag naar de kinderen van Bernhard kunnen we immers ook als volgt stellen:

```
| ?- ouder-van(bernhard,X).
```

Complexe vraagwoord-vragen kunnen worden gesteld door gebruik te maken van meerledige doelclausules.

```
| ?- moeder_van(juliana,X),
     ouder_van(X,Y).

X = beatrix,
Y = willem_alexander ;

no
| ?-
```

Deze vraag suggereert de definitie van het begrip *grootmoeder* met behulp van de volgende regel:

```
grootmoeder_van(X,Y) :-
   moeder_van(X,Z), ouder_van(Z,Y).
```

Terugvertaald naar standaard-notatie:

$$\forall x \forall y \forall z((Mxz \wedge Ozy) \rightarrow Gxy).$$

Dit is equivalent met:

$$\forall x \forall y (\exists z (Mxz \wedge Ozy) \rightarrow Gxy).$$

Je ziet: bij het programmeren in PROLOG komen de vaardigheden in het vertalen van Nederlands naar predikatenlogica die je in Hoofdstuk 11 aangeleerd goed van pas.

Na uitbreiding van het programma uit voorbeeld 16.1 met de nieuwe regel reageert het systeem als volgt op het bevragen van de grootmoeder-relatie:

```
| ?- grootmoeder_van(X,Y).

X = juliana,
Y = willem_alexander ;

no
| ?-
```

Het systeem gebruikt regels door hun kop te 'unificeren' met de gestelde vraag, en dan de zo ontstane nieuwe doelen te onderzoeken. Unificeren is gelijk maken door het vinden van een passende substitutie voor de variabelen. Het doel

```
| ?- grootmoeder_van(X,Y).
```

wordt geünificeerd met de kop van de regel

```
grootmoeder_van(X,Y) :-
   moeder_van(X,Z), ouder_van(Z,Y).
```

Dit levert de volgende nieuwe subdoelen:

```
?- moeder_van(X,Z), ouder_van(Z,Y).
```

Het eerste subdoel wordt geünificeerd met de regel

```
moeder_van(X,Y) :-
   vrouw(X), ouder_van(X,Y).
```

Dit levert de volgende nieuwe subdoelen:

```
?- vrouw(X), ouder_van(X,Z), ouder_van(Z,Y).
```

Vervolgens worden de feiten voor *vrouw* en *ouder* successievelijk uitgeprobeerd tot een substitutie voor de individuele variabelen gevonden is die aan alle condities voldoet.

Opgave 16.1 Breid het programma uit voorbeeld 16.1 uit met meer feiten in termen van `man`, `vrouw` en `ouder_van`, en met regels die de begrippen *broer*, *zus*, *oom* en *tante* definiëren.

Opgave 16.2 Breid het programma uit voorbeeld 16.1 uit met een predikaat `voorouder_van`.

Opgave 16.3 Programmeer de inhoud van de volgende onsterfelijke tekst: "Al die willen uit varen gaan, moeten mannen met baarden zijn. Jan, Piet, Joris en Korneel, die hebben baarden, zij varen mee." Gebruik predikaten `man`, `vrouw`, `bebaard` en `vaart_mee`. Hoe wordt de vraag naar wie er meevaart gesteld, en wat is het antwoord?

16.1 PROLOG met gestructureerde data

Zolang we alleen individuele constanten en variabelen gebruiken zal het domein waar we over praten eindig zijn: omdat een PROLOG programma eindig is kunnen er slechts eindig veel objecten bij name worden genoemd. PROLOG programma's worden interessanter wanneer we functieconstanten gebruiken om gestructureerde objecten weer te geven. De functieconstanten maken het mogelijk om een oneindig domein van objecten op te bouwen.

Voorbeeld 16.2 Dit programma illustreert het gebruik van functie-constanten:

```
:- op(600, fy, s).

plus(X, 0, X).
plus(X, s Y, s Z) :- plus(X, Y, Z).
```

De doelclausule met `op` is nodig om `s` te declareren als een prefix functie-constante, eenplaatsig en met 'precedentie' 600. Meer informatie over `op` is te vinden in elk PROLOG leerboek. We gebruiken `s` als naam voor de opvolger functie. De representatie voor de natuurlijke getallen wordt dus: $\{0, s0, ss0, sss0, \ldots\}$.

Het voorbeeldprogramma geeft een definitie van optelling voor de natuurlijke getallen. De relatie `plus` geldt tussen drie getallen wanneer optellen van het eerste getal bij het tweede getal het derde getal oplevert. Terugvertalen naar standaard-notatie is weer instructief (s staat voor de opvolger-functie):

$$\forall x Px0x \wedge \forall x \forall y \forall z (Pxyz \rightarrow Pxsysz).$$

Vragen hoeveel twee plus twee is gaat nu als volgt:

```
| ?- plus(s s 0, s s 0, X).

X = s s s s 0 ;

no
| ?-
```

Vragen naar alle mogelijkheden om door middel van optelling van natuurlijke getallen het getal *drie* te krijgen:

```
| ?- plus(X, Y, s s s 0).

X = s s s 0,
Y = 0 ;

X = s s 0,
Y = s 0 ;
```

```
X = s 0,
Y = s s 0 ;

X = 0,
Y = s s s 0 ;

no
| ?-
```

Voorbeeld 16.3 We kunnen het programma uit voorbeeld 16.2 verder uitbreiden door clausules voor vermenigvuldiging toe te voegen:

```
maal(_, 0, 0).

maal(X, s Y, U) :-
   maal(X, Y, Z), plus(X, Z, U).
```

In de eerste clausule staat _ voor een variabele. Variabelen in PROLOG beginnen met een hoofdletter of een _-teken. Het verschil tussen deze twee is dat _-variabelen uniek zijn in een clausule: ze spelen om zo te zeggen slechts een bijrol; het eigenlijke werk wordt verricht door variabelen die meerdere keren in één clausule voorkomen.

Het programma uit voorbeeld 16.3 kan als volgt worden bevraagd:

```
| ?- maal(X,X,Y).

X = Y = 0 ;

X = Y = s 0 ;

X = s s 0,
Y = s s s s 0 ;

X = s s s 0,
Y = s s s s s s s s s 0 ;

X = s s s s 0,
Y = s s s s s s s s s s s s s s s s 0
```

Het is duidelijk dat elke nieuwe ;–aansporing een nieuw antwoord oplevert. Dit gaat door totdat de gebruiker er genoeg van heeft of de zaak wordt afgebroken met een foutmelding (*'out of memory'* of iets dergelijks). Dit gedrag is natuurlijk correct: de gestelde vraag heeft nu eenmaal een oneindig aantal goede antwoorden. De vraagstelling suggereert meteen de definitie van een nieuw predikaat voor kwadrateren:

```
kwadr(X, Y) :- maal(X, X, Y).
```

Er zijn overigens ook voorbeelden waar PROLOG 'ten onrechte' in een lus terechtkomt:

Voorbeeld 16.4 Beschouw dit programma:

```
getrouwd_met(claus, beatrix).
getrouwd_met(X,Y) :- getrouwd_met(Y,X).
```

Logisch gesproken is één en ander correct: de regel drukt uit dat de *getrouwd zijn met* relatie symmetrisch is. Het PROLOG systeem verslikt zich hierin: geconfronteerd met de vraag `:- getrouwd_met(X,Y)` zal het eindeloos de antwoorden `X = claus, Y = beatrix` en `X = beatrix, Y = claus` blijven herhalen. Overigens is hier in dit geval iets aan te doen door preciezer formuleren. Het volgende programma vermijdt de vicieuze cirkel:

```
getrouwd_met_mv(claus, beatrix).
getrouwd_met_mv(bernhard, juliana).

getrouwd_met(X,Y) :-
   getrouwd_met_mv(X,Y).

getrouwd_met(X,Y) :-
   getrouwd_met_mv(Y,X).
```

Het voorbeeld illustreert dat de PROLOG programmeur niet alleen rekening moet houden met de logische betekenis van de programma-clausules maar ook met de manier waarop PROLOG het programma verwerkt. PROLOG programma's hebben niet alleen een *logische* of *declaratieve* interpretatie maar ook een *procedurele* interpretatie. De logische en de procedurele kijk leveren helaas niet altijd hetzelfde plaatje op.

Opgave 16.4 Breid de clausules uit voorbeelden 16.2 en 16.3 uit met een predikaat `fac(X,Y)` voor het uitrekenen van de faculteitsfunctie. De recursieve definitie van deze functie luidt als volgt:

– $0! = 1$;

– $(n+1)! = n! \cdot (n+1) \quad (n \geq 0)$.

Voorbeeld 16.5 De natuurlijke getallen worden in de programma's uit voorbeelden 16.2 en 16.3 gerepresenteerd als gestructureerde objecten. Een ander voorbeeld van het gebruik van functiesymbolen voor het aanduiden van gestructureerde objecten is het gebruik van functies voor het construeren van lijsten. Lijsten of eindige rijtjes kunnen worden opgebouwd uit de lege lijst, die we `nil` zullen dopen, en een verzameling van willekeurige objecten, met behulp van een constructie-functie **.** die een nieuw element voor aan een lijst plakt. Een BNF regel voor de opbouw van lijsten is

lijst ::= **nil** | *element* **.** *lijst*

en in PROLOG ziet dit er als volgt uit:

```
:- op(600, xfy, .).

lijst(nil).
lijst(_Kop.Staart) :- lijst(Staart).
```

Weer is de doelclausule met op bedoeld om een functiesymbool te declareren: tweeplaatsig, infix-notatie, rechts-vertakkend en met 'precedentie' 600. De PROLOG leerboeken geven meer informatie.

De vraag of a.b.c.nil een lijst is wordt als volgt gesteld en beantwoord:

```
| ?- lijst(a.b.c.nil).

yes
| ?-
```

Voorbeeld 16.6 De volgende uitbreiding van het programma in voorbeeld 16.5 maakt het mogelijk om te praten over de elementen van een lijst.

```
elem(X,X.Y) :- lijst(Y).
elem(X,_Kop.Staart) :- elem(X,Staart).
```

Een object is element van een lijst L als dat object (i) de kop is van lijst L of (ii) een element is de staart van lijst L.

Hier zie je het predikaat in actie:

```
| ?- elem(b,a.b.c.nil).

yes
| ?- elem(d,a.b.c.nil).

no
| ?- elem(X,a.b.c.nil).

X = a ;

X = b ;

X = c ;

no
| ?-
```

In feite heeft PROLOG een iets handiger notatie voor lijsten ingebakken. De lijst nil wordt genoteerd als [], de lijst a.b.c.nil als [a,b,c], en in het algemeen, de lijst met kop a en een willekeurige staart als [a|X] of [a|_]. De notatie met | en variabelen specificeert in feite *lijstpatronen*. Het patroon [a,b|_] specificeert een lijst met eerste element a, tweede element b, en een willekeurige staart. Let op: elementen van lijsten kunnen zelf weer lijsten zijn: [a] is een lijst met één element, te weten het object a, en [[a]|_] is het patroon van alle lijsten die met de lijst [a] beginnen. Merk op dat [a|[]] een andere aanduiding is voor de lijst [a].

Opgave 16.5 Omschrijf de volgende lijstpatronen:

1. [a,_,b|_].
2. [[a,_,b]|_].
3. [[]|_].
4. [[],_].
5. [_,_,a].

Het algemene patroon van een niet-lege lijst is: [_|_] of [Kop|Staart].

Voorbeeld 16.7 Door gebruik te maken van de PROLOG lijst notatie kan het bovenstaande lijstprogramma vervangen worden door de volgende clausules:

```
element(X,[X|_Staart]).

element(X,[_Kop|Staart]) :-
   element(X,Staart).
```

Het apart definiëren van het begrip *lijst* is nu overbodig geworden: het gebruik van de haken [en] dwingt af dat het tweede argument van element een lijst is.

Als we vragen wanneer a een element is van een lijst krijgen we de volgende antwoorden:

```
| ?- element(a,X).

X = [a|_224] ;

X = [_223,a|_226] ;

X = [_223,_225,a|_228] ;

X = [_223,_225,_227,a|_230]
```

Dit is wat er letterlijk op het scherm verschijnt; de getalsaanduidingen van de variabelen (_223, _224, enzovoorts) zijn 'toevallig'; het PROLOG systeem kiest deze aanduidingen op grond van de geheugenlocaties waar de variabelen zich bevinden.

Parafrase: a is element van X wanneer X een lijst is met a als eerste element, of een lijst met a als tweede element, of een lijst met a als derde element, of een lijst met a als vierde element, enzovoorts. Weer is het aantal mogelijke antwoorden oneindig. De antwoordenreeks kan worden afgebroken door een *'return'* te geven in plaats van een druk op de ';' toets.

Opgave 16.6 Welke lijst-eigenschap wordt gedefinieerd door het volgende predikaat:

```
xyz([]).
xyz([_,_|Staart]) :- xyz(Staart).
```

Opgave 16.7 Welke lijst-eigenschap wordt gedefinieerd door het volgende predikaat:

```
zyx(X,Y,[X,Y|_]).
zyx(X,Y,[_|Staart]) :- zyx(X,Y,Staart).
```

Voorbeeld 16.8 Twee lijsten aan elkaar plakken ('concateneren') gebeurt met het volgende programma:

```
concat([],X,X).

concat([Kop|Staart],X,[Kop|Y]) :-
   concat(Staart,X,Y).
```

Hier zie je het programma in actie:

```
| ?- concat([a,b],[c,d],X).

X = [a,b,c,d] ;

no
| ?-
```

De volgende interactie illustreert het gebruik van `concat` om te vragen naar het 'verschil' van twee lijsten:

```
| ?- concat([a],X,[a,b,c,d]).

X = [b,c,d] ;

no
| ?-
```

Ook kunnen we vragen naar alle mogelijke manieren om een lijst op te splitsen in een prefix en een suffix:

```
| ?- concat(X,Y,[a,b,c]).

X = [],
Y = [a,b,c] ;

X = [a],
Y = [b,c] ;

X = [a,b],
Y = [c] ;

X = [a,b,c],
Y = [] ;

no
| ?-
```

Opgave 16.8 Definieer een predikaat `laatste(X,Y)` dat waar is desda `X` het laatste element is van een niet-lege lijst `Y`.

Opgave 16.9 Definieer een predikaat `keerom(X,Y)` dat een lijst `X` omkeert, met het resultaat in `Y`. Met andere woorden: `keerom(X,Y)` is waar desda `Y` de omkering van lijst `X` bevat. Aanwijzing: begin met het opschrijven van een recursieve definitie van de omkeer-procedure en formuleer die definitie daarna in PROLOG.

De bovenstaande discussie is bedoeld om je een zeker gevoel bij te brengen voor programmeren in PROLOG. Goede bronnen voor meer informatie zijn [Bradko 1986] en [Pereira & Shieber 1987]; het laatste boek geeft toepassingen van PROLOG voor de analyse van natuurlijke taal.

16.2 De PROLOG bewijsstrategie

Zoals reeds vermeld hebben PROLOG programma's een declaratieve en een procedurele interpretatie. In de procedurele interpretatie is een programma-clausule die de vorm heeft van een regel een definitie van een *procedure*: om twee lijsten `X` en `Y` aan elkaar te plakken moet je eerst de staart van lijst `X` concateneren met lijst `Y`, en vervolgens de kop van lijst `X` voor het resultaat zetten. Een doelclausule kan procedureel worden opgevat als een opdracht om een procedure uit te voeren. Het lege doel $\square$ kan worden opgevat als de lege procedure, ofwel: de opdracht om niets te doen, dat wil zeggen de halt-opdracht.

Voor een goed begrip van PROLOG is inzicht in het bewijsproces dat eraan ten grondslag ligt van belang, want de logische en de procedurele interpretatie van PROLOG programma's gaan niet altijd gelijk op. Als PROLOG puur declaratief zou zijn zou de programmeur zich niet hoeven bekommeren om de 'geest in de machine'. De redeneerregel waar alles om draait is zogenaamde *resolutie met unificatie*. Eerst een voorbeeld van resolutie zonder unificatie. Neem aan dat A,B,C,D,E,F predikaatletters zonder variabelen zijn (nulplaatsige predikaatletters of predikaatletters gevolgd door constanten):

$$\frac{A \text{ :- } B,C,D \qquad C \text{ :- } E,F}{A \text{ :- } B,E,F,D.}$$

De resolutie vindt hier plaats op C. Omdat de predikaten geen variabelen bevatten zijn er geen substituties nodig.

Hier is een voorbeeld van resolutie met unificatie; om de resolutie te doen slagen moeten er nieuwe waarden voor de variabelen worden gesubstitueerd.

$$\frac{A(x,y) \text{ :- } B(x),C(x,fy) \qquad C(a,u) \text{ :- } D(u),E(gu)}{A(a,fy) \text{ :- } B(a),D(fy),E(gfy).}$$

De resolutie vindt plaats op C; de unificerende substitutie—dat wil zeggen de gelijktijdige substitutie die de atomaire formules waarop de resolutie plaats vindt aan elkaar gelijk maakt—is $\{a/x, fy/u\}$, of in PROLOG notatie: `X = a, U = fY`.

In een PROLOG systeem wordt de resolutie-met-unificatie techniek als volgt gebruikt. Het systeem tracht een tegenspraak af te leiden uit $\neg\exists x_1 \cdots \exists x_m (A_1 \wedge \cdots \wedge A_m)$, genoteerd als :- $A_1,\ldots,A_m$, en een verzameling programma-clausules Π. Daartoe wordt het eerste (meest linkse) subdoel geünificeerd met de bovenste in aanmerking komende clausule in de lijst van programma-clausules. PROLOG maakt daarbij essentieel gebruik van het feit dat de verzameling Π geordend is als een lijst. Resolutie levert vervolgens een nieuwe lijst van subdoelen op: wanneer de gebruikte programma-clausule een feit is zal de nieuwe lijst een item minder bevatten; wanneer het een regel is kan de lijst van subdoelen groeien. De gevonden waarden van de variabelen worden opgeslagen; in PROLOG jargon: de *bindingen* van de

variabelen blijven bewaard. Als er een vraagwoord-vraag is gesteld gaan die opgeslagen waarden de antwoorden vormen. Vervolgens wordt het nieuwe eerste subdoel vergeleken met de in aanmerking komende programma-clausules, weer van boven naar beneden, enzovoorts.

Wanneer dit proces vastloopt omdat een subdoel S niet kan worden geünificeerd gaat het systeem *terugkrabbelen* (de Engelse benaming voor dit proces is *backtracking*). Het systeem 'keert dan terug in de zoekruimte'. De laatst geselecteerde programma-clausule P (die S ergens in zijn staart heeft, en die gebruikt was voor de resolutie van doel D) wordt verworpen, het systeem maakt de substitutie die D oploste tegen de kop van P ongedaan en neemt D opnieuw in beschouwing door te proberen D te unificeren met een van de clausules die op P volgen. Wanneer dit proces uiteindelijk een lege doelclausule oplevert zeggen we dat het falsum $\square$ is afgeleid; hiermee is $\neg\exists x_1 \cdots \exists x_m (A_1 \wedge \cdots \wedge A_m)$ weerlegd, en de gevonden bindingen voor de variabelen leveren de antwoord-substituties.

Ook na het vinden van een weerlegging (het afleiden van de lege doelclausule $\square$) krabbelt het systeem terug om nieuwe antwoorden te vinden. De unificatie die het laatste subdoel S heeft weggewerkt door middel van resolutie tegen een feit P in de lijst van programma-clausules wordt ongedaan gemaakt, en er wordt gepoogd een substitutie te vinden die S unificeert met de kop van een van de programma-clausules die op P volgen. Dit kan leiden tot nieuwe antwoord-substituties, als $\square$ vanaf dit punt op andere manieren kan worden afgeleid, of tot verder terugkrabbelen op de wijze die in de vorige alinea is uiteengezet.

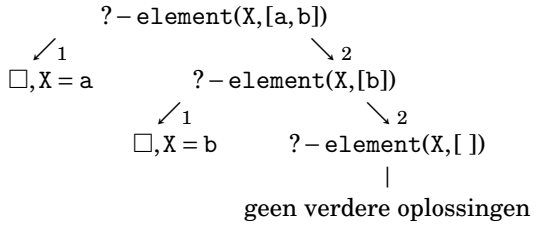
In het PROLOG zoek- en reken-algoritme zitten twee keuze-elementen, te weten:

– **PROLOG rekenregel:** pak altijd eerst het meest linkse subdoel in de doel-clausule aan.

– **PROLOG zoekregel:** doorzoek de lijst van programma-clausules altijd van boven naar beneden.

Tezamen levert dit een 'eerst links en omlaag' bewijsstrategie op. Voordeel van deze strategie is de efficiëntie. Nadeel is dat de bewijsstrategie niet *volledig* is. Er is geen garantie dat elke weerlegging wordt gevonden omdat het systeem terecht kan komen in een oneindig zoekpad 'links-omlaag' in de boom van mogelijke oplossingen, terwijl er rechts nog niet geëxploreerde eindige paden zijn die naar een oplossing leiden.

Het is illustratief om voor een gegeven voorbeeldprogramma en doelclausule de 'boom van mogelijke oplossingen' uit te tekenen. Neem weer het programma uit voorbeeld 16.7, en beschouw de vraag `?- element(X,[a,b])`. We nummeren de twee programma-clausules als 1 en 2. De bewijsboom ziet er nu zo uit (alleen bevraagde variabelen zijn in de boom opgenomen):

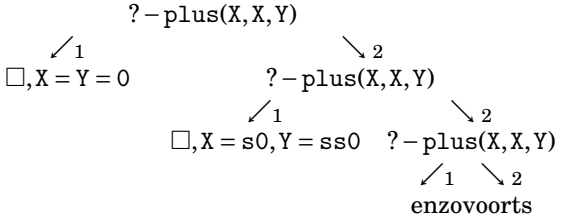


Deze boom wordt volgens de 'eerst links en naar beneden' strategie doorzocht. Merk op dat de bewijsboom voor dit voorbeeld eindig is: doorzoeken volgens een andere strategie (bij voorbeeld: 'eerst rechts en in de breedte') zou dezelfde antwoorden opleveren.

Beschouw nu het programma uit voorbeeld 16.2. We nummeren de twee programma-clausules weer als 1 en 2. De bewijsboom voor de vraag

```
?- plus(X,X,Y)
```

ziet er als volgt uit:

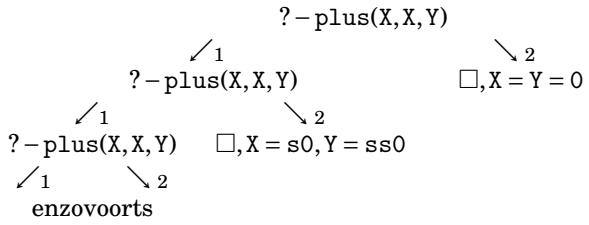


Deze bewijsboom heeft een oneindige tak; bij doorlopen van de boom volgens de PROLOG bewijsstrategie komen er oneindig veel antwoordsubstituties te voorschijn.

Veronderstel nu dat we de volgorde van de twee programma-clausules omkeren:

```
plus(X, s Y, s Z) :- plus(X, Y, Z).
plus(X, 0, X).
```

Weer nummeren we de twee clausules als 1 en 2. Een bewijsboom voor de vraag `?- plus(X,X,Y)` ziet er nu zo uit:



PROLOG zal deze boom weer volgens de 'eerst links en naar beneden' strategie doorlopen, en ... het systeem raakt daarbij in een oneindige lus waarbij geen antwoorden worden gegenereerd.

De moraal: bij het schrijven van PROLOG programma's en bij het bevragen van die programma's moeten we er voor zien te zorgen dat een eventuele oneindige tak in de bewijsboom voor dat programma met die vraag altijd de meest rechtse tak in de boom is. Vaak

lukt dit door ordening van de programma-clausules (feiten altijd vóór regels), maar vaak ook niet. In dit laatste geval is de enige uitweg het gebruik van de zogenaamde *snoei-operator* (Eng.: *cut operator*) om een gedeelte van de bewijsboom weg te snoeien. De snoei-operator is een zuiver procedureel werktuig: het is nodig om van PROLOG een volwassen procedurele programmeertaal te maken, maar het verstoort de illusie van PROLOG als puur declaratieve taal. Bespreking van het gebruik van de snoei-operator valt buiten het bestek van dit dictaat.

16.3 Modellen voor PROLOG programma's

We schakelen terug naar het puur logisch/declaratieve perspectief op PROLOG programma's voor een paar semantische vragen.

Definitie 16.3 *Het* **Herbrand universum** *van een PROLOG programma* Π*, notatie* H_Π*, is de verzameling van alle variabel-vrije termen in de taal van* Π*; als zulke termen niet bestaan omdat de taal van* Π *geen individuele constanten bevat dan nemen we een constante a als voorgift.*

Een eenvoudig Herbrand universum zag je al in voorbeeld 16.2: de verzameling $\mathbb{N}$. De volgende stelling is elementair maar cruciaal.

Stelling 16.1 *Elk PROLOG programma is consistent.*

Bewijs: Laat Π een verzameling programma-clausules zijn in de predikatenlogische taal $\mathcal{T}$. We construeren een model voor Π. We nemen het Herbrand universum H_Π als domein van het model. We interpreteren een individuele constante c als die constante zelf. Net zo: de interpretatie van een functie-constante f toegepast op een n-tal termen $t_1,\ldots,t_n$ is $f(t_1,\ldots,t_n)$ zelf. Neem voor predikaatletters de ruimst mogelijke interpretatie: voor 0-plaatsige predikaatletters (dat wil zeggen: propositieletters) is dat de waarde 1, voor 1-plaatsige de verzameling objecten H_Π, voor 2-plaatsige H_Π^2 en in het algemeen voor n-plaatsige predikaten de verzameling H_Π^n. Omdat elke atomaire formule in Π dan waar is, zijn ook alle programma-feiten (universele generalisaties van atomen) waar. Ook zijn alle conjuncties van atomen, implicaties van zulke conjuncties naar atomen, en universele generalisaties over zulke implicaties waar (ga na!), en daarmee zijn de programma-regels in Π waar. □

Modellen zoals in stelling 16.1 geconstrueerd staan bekend als *Herbrand modellen* (naar de Franse logicus Jacques Herbrand). Niet alle Herbrand modellen zijn even 'zuinig': de methode uit het bewijs van de stelling maakt elk n-plaatsig predikaat dat in Π voorkomt waar voor alle n-tallen van individuen in het Herbrand universum. Deze interpretatie maakt de clausules uit Π zeker waar, maar daarnaast nog veel meer. De volgende definitie maakt het mogelijk over te stappen op zuiniger Herbrand modellen.

Definitie 16.4 *Laat* $\{\mathcal{M}_i \mid i \in I\}$ *een familie van predikatenlogische structuren zijn voor taal* $\mathcal{T}$ *met allemaal hetzelfde domein en dezelfde operaties. De* **doorsnede** *van deze familie, notatie* $\bigcap_{i\in I}\mathcal{M}_i$*, is de structuur met hetzelfde domein en dezelfde operaties, en met de interpretatie* I *van een* n*-plaatsige predikaatletter* P *als volgt gedefinieerd:*

$$\langle d_1,\ldots,d_n\rangle \in I(P)$$

desda de interpretatie van P *waar is van* $\langle d_1,\ldots,d_n\rangle$ *in elke* $\mathcal{M}_i$*.*

Stelling 16.2 *Als een PROLOG programma* Π *waar is in een familie*

$$\{\mathcal{M}_i \mid i \in I\}$$

van predikatenlogische structuren voor taal $\mathcal{T}$ *van* Π *met allemaal hetzelfde domein en dezelfde operaties, dan is* Π *waar in de doorsnede van deze familie* $\bigcap_{i\in I}\mathcal{M}_i$*.*

Bewijs: Neem een willekeurige $\varphi \in \Pi$. Als φ van de vorm $\forall x_1 \cdots \forall x_m A$ is, met A atomair, en φ is waar in elk van de $\mathcal{M}_i$, dan volgt het direct uit de definitie van 'doorsnede' voor structuren dat φ waar is in $\bigcap_{i\in I}\mathcal{M}_i$. Veronderstel nu dat φ de vorm heeft van een regel:

$$\forall x_1 \cdots \forall x_m((A_1 \wedge \cdots \wedge A_n) \rightarrow B).$$

Neem aan dat er een bedeling b is zo dat

$$\bigcap_{i\in I}\mathcal{M}_i \models A_1 \wedge \cdots \wedge A_n \ [b].$$

Dan vervult b elke A_j $(1 \le j \le n)$ in elk van de $\mathcal{M}_i$ $(i \in I)$. Omdat gegeven is dat $\mathcal{M}_i \models \varphi$ mogen we concluderen dat b het predikaat B vervult in $\mathcal{M}_i$, voor elke $i \in I$. Dus:

$$\bigcap_{i\in I}\mathcal{M}_i \models B \ [b].$$

Hieruit volgt dat $\bigcap_{i\in I}\mathcal{M}_i \models \varphi$. □

Stelling 16.2 geeft ons een methode om een kleinste Herbrand model voor een programma Π te construeren.

Definitie 16.5 *Het* **kleinste Herbrand model** *van een programma* Π *is de doorsnede van alle Herbrand modellen van* Π*.*

Een Herbrand model voor het programma uit voorbeeld 16.1 heeft het volgende individuendomein: `willem_alexander`, `claus`, `bernhard`, `beatrix` en `juliana`. Een mogelijke interpretatie die alle programma-clausules waar maakt is: elk individu is een man, elk individu is een vrouw, elk individu is ouder van iedereen. Dit is niet de meest zuinige interpretatie. In het kleinste Herbrand model voor het programma zijn `willem_alexander`, `claus` en `bernhard` de mannen, `beatrix` en `juliana` de vrouwen, en is de relatie `ouder_van` beperkt tot

$$\{\langle \texttt{juliana},\texttt{beatrix}\rangle,\ \langle \texttt{bernhard},\texttt{beatrix}\rangle,\ \langle \texttt{beatrix},\texttt{willem\_alexander}\rangle,\ \langle \texttt{claus},\texttt{willem\_alexander}\rangle\}.$$

Alle Herbrand modellen voor het rekenprogramma uit voorbeeld 16.2 en 16.3 hebben het volgende aftelbaar oneindige domein:

$$\{0, s0, ss0, sss0, \ldots\}.$$

Het grootste van deze modellen maakt *plus* en *maal* waar voor alle drietallen van termen (dat wil zeggen, objecten in het domein van het model). Er zijn ook allerlei zuiniger mogelijkheden. De doorsnede constructie gooit al het overbodige overboord en geeft ons precies wat we willen: de natuurlijke getallen met de standaard interpretatie van *plus* en *maal*.

Herbrand modellen voor het programma uit voorbeeld 16.7 hebben als domein de verzameling van alle lijsten die zijn opgebouwd uit `[]` (immers, `[]` is een constante). Dus: `[]`, `[[]]`, `[[],[]]`, enzovoorts. In het kleinste Herbrand model geldt de *element* relatie van precies die paren met de eigenschap dat de eerste lijst een element is van de tweede.

Om iets meer te kunnen zeggen over kleinste Herbrand modellen hebben we een definitie en een tweetal hulpstellingen nodig.

Definitie 16.6 *Een predikatenlogisch model $\mathscr{M}_1 = \langle D_1, I_1\rangle$ is een* **substructuur** *van een model $\mathscr{M}_2 = \langle D_2, I_2\rangle$, notatie $\mathscr{M}_1 \subseteq \mathscr{M}_2$, als het volgende geldt:*

- *$D_1 \subseteq D_2$;*
- *op D_1 stemmen de $\mathscr{M}_2$ functies overeen met de $\mathscr{M}_1$ functies, en D_1 is afgesloten onder die functies;*
- *de interpretaties van individuele constanten worden gezien als 0-plaatsige functies: al die interpretaties bevinden zich in D_1;*
- *op D_1 stemmen de $\mathscr{M}_2$ relaties overeen met die van $\mathscr{M}_1$.*

Intuïtief gesproken is een substructuur van een model voor een taal $\mathscr{T}$ een model voor $\mathscr{T}$ dat mogelijkerwijs een kleiner domein heeft, maar dat zich verder zoveel mogelijk gedraagt als het oorspronkelijke model.

Voorbeelden: de natuurlijke getallen met nul, de opvolger functie en de kleiner dan relatie vormen een substructuur van de reële getallen met nul, opvolger en kleiner dan. Net zo: de even natuurlijke getallen met nul en de kleiner dan relatie vormen een substructuur van de natuurlijke getallen met nul en kleiner dan. In dit laatste geval is het niet mogelijk de opvolger functie toe te voegen, want de even natuurlijke getallen zijn daaronder niet afgesloten.

Opgave 16.10 Leg uit waarom de natuurlijke getallen met nul en de operaties voor optellen en aftrekken geen substructuur vormen van de gehele getallen met nul en de operaties voor optellen en aftrekken.

Stelling 16.3 *Voor elke universele predikatenlogische zin φ, dat wil zeggen voor elke zin φ die equivalent met een formule bestaande uit een rijtje van universele kwantoren gevolgd door een kwantorvrij deel, geldt: als $\mathscr{M}_1 \subseteq \mathscr{M}_2$ en $\mathscr{M}_2 \models \varphi$, dan is $\mathscr{M}_1 \models \varphi$.*

Bewijs: We mogen aannemen dat φ de vorm $\forall x_1 \cdots \forall x_m \psi(x_1, \ldots, x_m)$ heeft, waarbij $\psi(x_1, \ldots, x_m)$ staat voor een kwantorvrije formule waarin de variabelen $x_1, \ldots, x_m$ vrij voorkomen. Neem aan dat $\mathscr{M}_2 \models \varphi$. Dan geldt voor elke bedeling b voor $x_1, \ldots, x_m$ in D_2 dat

$$\mathscr{M}_2 \models \psi(x_1, \ldots, x_m)\,[b].$$

Maar dan vervult ook elke bedeling b' die aan alle variabelen objecten uit D_1 toebedeelt $\psi(x_1, \ldots, x_m)$ in $\mathscr{M}_2$. En omdat $\mathscr{M}_1$ en $\mathscr{M}_2$ op het domein van $\mathscr{M}_1$ met elkaar overeenstemmen wat de interpretatie van functie-constanten en predikaatletters betreft hebben we, voor elke bedeling b' voor $\mathscr{M}_1$:

$$\mathscr{M}_1 \models \psi(x_1, \ldots, x_m)\,[b'].$$

Met andere woorden: $\mathscr{M}_1 \models \varphi$. □

Stelling 16.4 *Elke consistente verzameling Γ van universele zinnen heeft een Herbrand model $\mathscr{H}$.*

Bewijs: Omdat Γ consistent is is er een $\mathscr{M} = \langle D, I\rangle$ met $\mathscr{M} \models \Gamma$. We construeren op basis van $\mathscr{M}$ een Herbrand model voor Γ, als volgt. De functie W die de variabelvrije termen van de taal afbeeldt op de waarden die ze volgens de interpretatiefunctie I van $\mathscr{M}$ moeten krijgen, beeldt het Herbrand universum H_Γ af in het domein van $\mathscr{M}$. Als f een m-plaatsige functieconstante is die in Γ wordt gebruikt, dan geldt het volgende:

$$W(f(t_1, \ldots, t_m)) = W(f)(W(t_1), \ldots, W(t_m)).$$

Met andere woorden: W beeldt elke complexe variabelvrije term af op het 'juiste' object in het domein van $\mathscr{M}$.

De definitie van $\mathscr{H}$ is nu simpel. Voor elke predikaatletter P die in Γ wordt gebruikt stipuleren we dat $P(t_1, \ldots, t_n)$ waar is in $\mathscr{H}$ desda

$$\langle I(t_1), \ldots, I(t_n)\rangle \in I(P).$$

Dit legt het Herbrand model $\mathscr{H}$ volledig vast.

We moeten nu nog laten zien dat $\mathscr{H} \models \Gamma$. Omdat alle formules uit Γ universele predikatenlogische zinnen zijn blijft de waarheid ervan bewaard onder het vormen van substructuren (stelling 16.3). Het beeld van H_Γ onder W is een substructuur van $\mathscr{M}$. Alle leden van Γ zijn dus waar in deze substructuur. Uit de definitie van de interpretatiefunctie voor $\mathscr{H}$ volgt nu dat $\mathscr{H} \models \Gamma$. □

De stelling waarmee we deze paragraaf besluiten illustreert het belang van het kleinste Herbrand model van een PROLOG programma Π.

Stelling 16.5 *Voor elke atomaire formule φ en elk PROLOG programma Π geldt: $\Pi \models \varphi$ desda φ waar is in het kleinste Herbrand model voor Π.*

Bewijs: Als $\Pi \models \varphi$ dan is φ zeker waar in het kleinste Herbrand model voor Π, want dit model maakt alle clausules uit Π waar.

Om het omgekeerde te laten zien veronderstellen we dat $\Pi \not\models \varphi$. Dit wil zeggen: er is een model $\mathscr{M}$ voor Π met $\mathscr{M} \models \neg\varphi$. We gebruiken nu de methode van stelling 16.4 om op basis van $\mathscr{M}$ een Herbrand model $\mathscr{H}$ te construeren. Uit de constructie volgt dat $\mathscr{H} \models \neg\varphi$, dat wil zeggen: $\mathscr{H}$ maakt φ niet waar. Maar dan zal het kleinste Herbrand model voor Π (dat alleen de atomaire formules waar maakt die in *elk* Herbrand model voor Π waar zijn), φ onwaar maken. □

Deze stelling rechtvaardigt een alternatieve definitie van het kleinste Herbrand model van een PROLOG programma Π als het Herbrand model waarvoor de interpretatie van elke predikaatletter P als volgt gegeven is:

$$\langle t_1,\ldots,t_n\rangle \in I(P) \quad \text{desda} \quad \Pi \models P(t_1,\ldots,t_n).$$

Dit model is bevat in alle Herbrand modellen voor Π.

Het is niet al te moeilijk om te zien dat stelling 16.5 niet alleen opgaat voor atomaire formules, maar ook voor conjuncties van atomaire formules en voor existentiële kwantificaties over zulke conjuncties. Daarmee is onze cirkel rond. We hebben immers in § 16 gezien dat doelclausules in PROLOG de volgende algemene vorm hebben:

$$\neg\exists x_1 \cdots \exists x_m (A_1 \wedge \cdots \wedge A_n).$$

PROLOG probeert zo'n doel te weerleggen, met andere woorden: als de weerleggingspoging slaagt is de PROLOG conclusie van de volgende algemene vorm:

$$\exists x_1 \cdots \exists x_m (A_1 \wedge \cdots \wedge A_n).$$

Nog anders gezegd: PROLOG conclusies hebben de vorm van atomaire formules, conjuncties van atomaire formules, of existentiële kwantificaties over atomaire formules of hun conjuncties. We hebben in stelling 16.5 dus bewezen dat voor PROLOG conclusies de twee begrippen *logisch volgen uit programma* Π en *waar zijn in het kleinste Herbrand model van* Π op hetzelfde neerkomen.

Hopelijk heeft deze paragraaf je een indruk kunnen geven van het theoretische fundament waarop PROLOG rust. Voor meer informatie over de theoretische achtergronden van programmeren met logica verwijzen we naar [Apt 1988].

Hoofdstuk 17

Programma-correctheid

Omdat PROLOG programma's een declaratieve lezing hebben wordt programmeren in PROLOG wel aangeduid als *declaratief programmeren*: een programma is in feite een verzameling *definities*. In meer traditionele programmeertalen is een programma een verzameling *opdrachten*; programmeren op deze meer traditionele manier wordt *imperatief programmeren* genoemd. Met name voor het onderzoek naar correctheid van (imperatieve) programma's is de predikatenlogica van groot belang.

17.1 Impertaal

Om het gebruik van predikatenlogica bij het leveren van correctheidsbewijzen voor imperatieve programma's te demonstreren, definiëren we een heel simpele imperatieve programmeertaal in BNF notatie. Voor het gemak breiden we de BNF notatie uit met accolades. We spreken af dat $\{A\}$ betekent dat A 0 of meer malen mag voorkomen. Deze uitgebreide BNF notatie wordt EBNF ('Extended Backus Naur Form') genoemd. De definitie van onze programmeertaal ziet er nu zo uit (elke opdracht in de taal kan worden beschouwd als een programma):

Definitie 17.1 (Impertaal) *De programmeertaal* **Impertaal**.

⟨*opdracht*⟩ ::=
⟨*variable*⟩ **:=** ⟨*term*⟩
| **begin** ⟨*opdracht*⟩ *{* **;** ⟨*opdracht*⟩ *}* **einde**
| **(** **als** ⟨*uitdrukking*⟩ **dan** ⟨*opdracht*⟩ **)**
| **(** **als** ⟨*uitdrukking*⟩ **dan** ⟨*opdracht*⟩ **anders** ⟨*opdracht*⟩ **)**
| **zolang** ⟨*uitdrukking*⟩ **doe** ⟨*opdracht*⟩

⟨*variabele*⟩ ::= ⟨*letter*⟩ | ⟨*letter*⟩ ⟨*getal*⟩

⟨*letter*⟩ ::= **a** | **b** | ⋯ | **x** | **y** | **z**

⟨*getal*⟩ ::= ⟨*cijfer*⟩ *{* ⟨*cijfer*⟩ *}*

⟨*cijfer*⟩ ::= **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9**

⟨*term*⟩ ::= ⟨*variable*⟩ | ⟨*getal*⟩ | **–** ⟨*getal*⟩
| ⟨*term*⟩ **+** ⟨*term*⟩ | ⟨*term*⟩ **–** ⟨*term*⟩
| **(** ⟨*term*⟩ ∗ ⟨*term*⟩ **)**

⟨*uitdrukking*⟩ ::= ⟨*term*⟩ = ⟨*term*⟩ | ⟨*term*⟩ < ⟨*term*⟩
| ⟨*term*⟩ > ⟨*term*⟩ | ⟨*term*⟩ <= ⟨*term*⟩
| ⟨*term*⟩ >= ⟨*term*⟩
| **niet** ⟨*uitdrukking*⟩
| **(** ⟨*uitdrukking*⟩ **en** ⟨*uitdrukking*⟩ **)**
| **(** ⟨*uitdrukking*⟩ **of** ⟨*uitdrukking*⟩ **)**

De tweeplaatsige term-operatoren <= en >= staan respectievelijk voor 'kleiner dan of gelijk aan' en 'groter dan of gelijk aan'. Ga na hoe de accolade-notatie die we boven hebben ingevoerd wordt gebruikt bij het herschrijven van ⟨*opdracht*⟩ en ⟨*getal*⟩.

Opgave 17.1 Geef een BNF definitie van ⟨getal⟩ die geen gebruik maakt van accolade-notatie.

Hier zijn een paar voorbeelden van programma's in **Impertaal**.

```
– x := x + 1
– (als x < 0 dan x := -x)
– (als x > y dan m := x anders m := y)
– begin z := x; x := y; y := z einde
– begin x := y; y := x einde
– begin
    i := 0;
    p := 0;
    zolang i < x doe
      begin p := p + y; i := i + 1 einde
  einde
– begin
    i := 0;
    f := 1;
    zolang i < x doe
      begin i := i + 1; f := (i * f) einde
  einde
```

De taal **Impertaal** (voor: 'imperatieve taal') die we hebben gedefinieerd is een sterk vereenvoudigde versie van Java. Variabelen en hun typen hoeven niet te worden gedeclareerd (in feite zijn alle variabelen van het type *geheel getal*). Merk op dat de BNF regels een dubbelzinnigheid uit de definitie van Java vermijden. De BNF regels voor Java maken niet duidelijk hoe de opdracht

if U_1 **then if** U_2 **then** O_1 **else** O_2

moet worden gelezen. De twee mogelijkheden zijn:

if U_1 **then (** **if** U_2 **then** O_1 **else** O_2**)**

en

if U_1 **then (** **if** U_2 **then** O_1**) else** O_2.

In **Impertaal** treedt de dubbelzinnigheid niet op, omdat de desambiguerende ronde haken verplicht aanwezig zijn.

17.2 Semantiek

We hebben nu de syntaxis van een programmeertaal. Wat we nog moeten vastleggen is wat het *effect* is dat onze programma's hebben. Met andere woorden: we moeten de semantiek van **Impertaal** nog specificeren. Daarvoor gaan we, in navolging van C.A.R. Hoare, predikatenlogica gebruiken. Omdat **Impertaal** zo simpel is, is het enige effect dat een programma kan hebben een verandering van de *toestand* van de machine waarop het programma wordt uitgevoerd. Een machinetoestand is niets anders is dan een specificatie van de waarden die de programma-variabelen hebben. Als bepaalde geheugenplaatsen waar variabelen zijn opgeslagen gevuld zijn met zekere waarden dan is het effect van een programma de verandering die dat programma teweeg brengt in die geheugenplaatsen.

Neem bij voorbeeld het programma `x := x + 1`. Dit programma bestaat uit een enkele *toekenningsopdracht*, een opdracht om een nieuwe waarde toe te kennen aan een variabele. In dit geval wordt de waarde van het getal dat in geheugenlocatie x zit opgeslagen met 1 verhoogd. Wanneer de geheugenlocatie x (de plaats in de computer waarin de waarde voor de variabele x is opgeslagen) voordat het programma gedraaid wordt de waarde 3 heeft, dan moet die locatie na afloop van het programma de waarde 4 hebben. Wat het programma doet is deze toestandsverandering teweeg brengen.

Opgave 17.2 Ga na welk effect de bovenstaande voorbeeldprogramma's hebben op de waarden van de variabelen die erin worden gebruikt.

Pre- en postcondities

Bij het uitvoeren van de bovenstaande opdracht heb je in feite een informele specificatie gegeven van de semantiek van de voorbeeldprogramma's. We gaan dat nu formeler maken: we gaan de voorwaarden waaraan een machinetoestand voldoet uitdrukken met behulp van predikatenlogische formules. We onderscheiden daarbij tussen *beginvoorwaarden* en *eindvoorwaarden*, of in jargon: tussen *pre-* en *post-condities*. Het tripel

(postconditie, programma, postconditie)

wordt genoteerd als een zogenaamd **Hoare tripel**:

$$\{\varphi\}\ \pi\ \{\psi\}.$$

Hierbij is π een programma in **Impertaal** (of een andere imperatieve taal waarover we correctheidsbeweringen willen doen), en zijn φ en ψ respectievelijk een beginvoorwaarde en een eindvoorwaarde voor de variabelen die in het programma gebruikt worden. Let op: de correctheidsbewering $\{\varphi\}\ \pi\ \{\psi\}$ garandeert niet dat het programma π afloopt. Nagaan of een programma termineert (dat wil zeggen: niet in een oneindige lus raakt) is een probleem op zichzelf. Dit probleem is veel moeilijker dan het op het eerste gezicht lijkt, en daarom laten stellen de beschouwing er van uit tot Sectie 17.5.

Omdat onze programmaspecificaties op zichzelf niet garanderen dat de programma's termineren wordt de correctheidsbewering $\{\varphi\}\ \pi\ \{\psi\}$ wel een *partiële correctheidsbewering* genoemd. De parafrase van $\{\varphi\}\ \pi\ \{\psi\}$ luidt:

> Als π een programma is dat termineert, en φ is een conditie die geldt voordat het programma wordt gedraaid, dan geldt conditie ψ wanneer het programma is afgelopen.

Voorbeeld 17.1 $\{x = 3\}$ `x := x + 1` $\{x = 4\}$.

Deze specificatie is juist: het is inderdaad het geval dat wanneer variabele x voor het draaien van programma `x := x + 1` de waarde 3 bevat, die variabele na afloop van het programma de waarde 4 zal bevatten.

Voorbeeld 17.2

$\{x = X, y = Y\}$
```
  begin z := x; x := y; y := z einde
```
$\{x = Y, y = X\}$.

Dit programma termineert, en het is niet moeilijk om in te zien dat het de waarden van x en y verwisselt. Om dit effect aan te kunnen geven ongeacht de beginwaarden van x en y gebruiken we hoofdletters X en Y voor die beginwaarden. De letters X en Y worden *schaduwvariabelen* genoemd (Eng.: *ghost variables*).

Voorbeeld 17.3

$\{x = X, y = Y\}$
```
  begin x := y; y := x einde
```
$\{x = Y, y = X\}$.

Deze specificatie is *niet* juist (ga na waarom).

Opgave 17.3 Vervang de begin- en eindvoorwaarden in dit voorbeeld door voorwaarden die juist zijn.

Voorbeeld 17.4 $\{\top\}\ \pi\ \{\psi\}$.

We gebruiken $\top$ voor een conditie die altijd waar is. De specificatie zegt dat ψ waar is als π termineert.

Opgave 17.4 Geef aan onder welke voorwaarden de specificatie $\{\varphi\}\ \pi\ \{\top\}$ juist is.

Voorbeeld 17.5

$\{\top\}$
```
    begin
      x := 0;
      zolang x >= 0 doe x := x + 1
    einde
```
$\{x = 100000\}$.

Deze specificatie is correct. Immers, het programma termineert niet, dus de bewering dat ALS het programma termineert de eindwaarde van x gelijk zal zijn aan 100000, is juist. Het voorbeeld maakt duidelijk dat we er voortdurend op bedacht moeten zijn dat de specificaties *partiële correctheidsbeweringen* geven.

Voorbeeld 17.6

```
{x ≥ 0}
   begin
     i := 0;
     p := 0;
     zolang i < x doe
       begin p := p + y; i := i + 1 einde
   einde
{p = x · y}.
```

De specificatie zegt dat als de waarde van x voordat het programma begint niet negatief is en het programma termineert, de variabele p na afloop het product van x en y zal bevatten. Ga na dat deze specificatie juist is.

Opgave 17.5 Is de specificatie nog steeds correct wanneer we de beginvoorwaarde vervangen door $\{\top\}$? Waarom (niet)?

We kunnen de probleemstelling ook omkeren: als een beginvoorwaarde en een eindvoorwaarde gegeven zijn, schrijf dan een programma in **Impertaal** dat aan deze specificatie voldoet. Zie de nu volgende opdrachten.

Opgave 17.6 Laat de volgende specificatie gegeven zijn:

$$\{x > 0\}\ \pi\ \{y = 1 + 2 + \cdots + x\}.$$

Geef een programma π dat gebruik maakt van de variabelen x en y, en dat aan deze specificatie voldoet.

Opgave 17.7 Laat de volgende specificatie gegeven zijn:

$$\{x > 0\}\ \pi\ \{y = 1^3 + 2^3 + \cdots + x^3\}.$$

Geef een programma π dat aan deze specificatie voldoet.

Formele semantiek

De betekenis van een programma π is formeel gedefinieerd als een relatie T tussen begin- en eindtoestanden (vulling van variabelen-registers).

Definitie 17.2 (Semantiek van Impertaal) *Laat π een programma zijn. De verzameling eind-toestanden waarin π ons kan brengen na te zijn gestart in een bepaalde begintoestand b wordt genoteerd met*

$$T(b,\pi).$$

Verder wordt gedefinieerd:

- $b' \in T(b, x := t) \Leftrightarrow b' = b(x|W_{M,b}(t))$.
- $b' \in T(b_1, \pi_1;\pi_2) \Leftrightarrow$ *er is een tussentoestand, b'', zó dat*

$$b'' \in T(b,\pi_1)\ \textit{en}\ b' \in T(b'',\pi_2).$$

- $b' \in T(b,$ ***als*** σ ***dan*** π_1 ***anders*** $\pi_2) \Leftrightarrow$

$$\begin{cases} b' \in T(b,\pi_1) & \textit{als } \mathscr{M}, b \models \sigma, \\ b' \in T(b,\pi_2) & \textit{anders.} \end{cases}$$

- $b' \in T(b,$ ***zolang*** σ ***doe*** $\pi) \Leftrightarrow$ *er is een eindig rijtje toestanden $b_1,\ldots,b_n$, zó dat $b_1 = b$, $b_n = b'$,*

$$b_{i+1} \in T(b_i,\pi),$$

$\mathscr{M}, b_i \models \sigma$ voor $i < n$ en $\mathscr{M}, b_n \not\models \sigma$.

Opmerkingen:

1. Omdat **Impertaal** deterministisch is, geldt altijd $|T(b,\pi)| \leq 1$.
2. Het gaat in deze definitie om de verzameling **eind**-toestanden waarin π ons kan brengen. Tussentoestanden zijn niet interessant.
3. Voor de betekenis van $b(x|W_{M,b}(t))$, zie Def. 11.1 op blz. 128.

De formulering van de betekenis van de assignment-statement is misschien wat intimiderend maar sluit waarschijnlijk geheel aan bij je intuïtie. Als bijvoorbeeld $t = y * (z-4)$, en $b(y) = 2$ en $b(z) = 7$, dan geldt

$$W_{M,b}(t) = W_{M,b}(y * (z-4)) = 2 * (7-4) = 6.$$

Volgens de definitie zou nu moeten gelden $b' \in T(b, x := y * (z-4)) \Leftrightarrow b' = b(x|6)$, met andere woorden, b' is gelijk aan b behoudens op coördinaat x, waar b' anders is vanwege $x := y * (z-4)$.

Definitie 17.3 (Correctheid van Hoare tripels) *Een Hoare tripel $\{\varphi\}\,\pi\,\{\psi\}$ heet* geldig *of* correct *in $\mathscr{M}$, notatie*

$$\mathscr{M} \models \{\varphi\}\,\pi\,\{\psi\},$$

als en slechts als voor alle $b_2 \in T(b_1,\pi)$ geldt: als $\mathscr{M}, b_1 \models \varphi$ dan $\mathscr{M}, b_2 \models \psi$.

In de logica wordt over geldigheid gesproken, en in de theorie van programmacorrectheid wordt over correctheid gesproken. Beide noties komen op hetzelfde neer.

Opgave 17.8 1. Voor welke programma-constructies geldt met zekerheid $|T(b,\pi)| = 1$?

2. Welke conclusie kun je trekken uit $|T(b,\pi)| = 1$? Uit $|T(b,\pi)| = 0$?

3. Beargumenteer dat, voor willekeurige φ, ψ en π de Hoare expressie $\mathscr{M} \models \{\varphi\}\,\pi\,\{\psi\}$ altijd geldig is als π nooit stopt in situaties waarbij $\mathscr{M}, b \models \varphi$.

   (Kom je er niet uit, denk dan aan het verschil tussen tussen-toestanden en eind-toestanden.)

We gaan wat voorbeeldjes doen.

Voorbeeld 17.7 Bewijs de geldigheid van het volgende Hoare tripel

$$\{x \le 2\} \;\; y := x+3 \;\; \{x+y \le 7\}.$$

Dat gaat zo. Zij $\mathscr{M}$ willekeurig. Stel

$$b_2 \in T(b_1, y := x+3) \tag{1}$$

voor willekeurige start-toestand b_1 met

$$\mathscr{M}, b_1 \models x \le 2. \tag{2}$$

en eind-toestand b_2. We moeten nu aantonen dat $\mathscr{M}, b_2 \models x+y \le 7$.

Vanwege Definitie 17.2 en (1) geldt dat

$$b_2 = b_1(y|b_1(x)+I(3)).$$

Vanwege (2) geldt dan

$$b_2(y) = b_1(x)+I(3) \le 5.$$

Verder blijft $b_2(x) = b_1(x) \le 2$.

Nu $\mathscr{M}, b_2 \models x+y \le 5 \Leftrightarrow b_2(x)+b_2(y) \le 7$, en we hebben al gecheckt dat dat laatste waar is.

Voorbeeld 17.8 Bewijs de geldigheid van het volgende Hoare tripel

$$\{x = 0\} \;\; \textbf{zolang } x < 5 \textbf{ doe } x := x+1 \;\; \{x = 5\}.$$

Zij $\mathscr{M}$ willekeurig. Stel

$$b' \in T(b, \textbf{zolang } x < 5 \textbf{ doe } x := x+1) \tag{3}$$

voor willekeurige start-toestand b met

$$\mathscr{M}, b \models x = 0. \tag{4}$$

en eind-toestand b'. We moeten nu aantonen dat $\mathscr{M}, b' \models x = 5$. Vanwege Definitie 17.2 en (3) geldt dat er een eindig rijtje toestanden $b_1, \ldots, b_n$ is, zó dat $b_1 = b$, $b_n = b'$,

$$b_{i+1} \in T(b_i, x := x+1), \tag{5}$$

$$\mathscr{M}, b_i \models x < 5 \text{ voor } i < n \text{ en } \mathscr{M}, b_n \not\models x < 5. \tag{6}$$

Vanwege (4) geldt $b_1(x) = 0$. Vanwege (5) geldt $b_{i+1} = b_i(x|b_i(x)+I(1))$ voor $i < n$. Dus $b_1(x) = 0$ en $b_{i+1}(x) = b_i(x)+1$ voor $i < n$. Daaruit volgt (officieel met volledige inductie) dat

$$b_i(x) = i \text{ voor } i \le n. \tag{7}$$

(Let op het "≤" teken.) Uit (6) volgt

$$b_i(x) < 5 \text{ voor } i < n \text{ en } b_i(x) \not< 5 \text{ voor } i = n. \tag{8}$$

Uit (7) en (8) volgt dat $b_n(x) = 5$. (Immers, de $b_i(x)$ loopt per iteratie één op.) Omdat $b' = b_n$ volgt $\mathscr{M}, b' \models x = 5$.

[1] Gekunstelde vertaling van het Engelse *vacuously*.

Voorbeeld 17.9 Bewijs de geldigheid van het volgende Hoare tripel

$$\{x = 1\} \;\; \textbf{zolang } x > 0 \textbf{ doe } x := x+1 \;\; \{x = -99\}.$$

Zij $\mathscr{M}$ willekeurig. Stel

$$b' \in T(b, \textbf{zolang } x > 0 \textbf{ doe } x := x+1) \tag{9}$$

voor willekeurige start-toestand b met

$$\mathscr{M}, b \models x = 1. \tag{10}$$

en eind-toestand b'. We gaan beargumenteren dat zo'n b' niet bestaat. Stel toch. (We hopen op een tegenspraak uit te komen.) Vanwege Definitie 17.2 en (9) zou dan volgen dat er een eindig rijtje toestanden $b_1, \ldots, b_n$ bestaat, zó dat $b_1 = b$, $b_n = b'$,

$$b_{i+1} \in T(b_i, x := x+1), \tag{11}$$

$$\mathscr{M}, b_i \models x > 0 \text{ voor } i < n \text{ en } \mathscr{M}, b_n \not\models x > 0. \tag{12}$$

Vanwege (10) geldt $b_1(x) = 1$. Vanwege (11) geldt $b_{i+1} = b_i(x|b_i(x)+I(1))$ voor $i < n$. Dus $b_1(x) = 1$ en $b_{i+1}(x) = b_i(x)+1$ voor $i < n$. Daaruit volgt (officieel met volledige inductie) dat

$$b_i(x) = i+1 \text{ voor } i \le n \tag{13}$$

Uit (12) volgt

$$b_i(x) > 0 \text{ voor } i < n \text{ en } b_i(x) \not> 0 \text{ voor } i = n. \tag{14}$$

Nu zijn (13) en (14) met elkaar in tegenspraak, immers de $b_i(x)$ loopt alleen maar op. Dus onze eerdere aanname dat er een b' bestaat zó dat (9) is onwaar. Dus

$$T(b, \textbf{zolang } x > 0 \textbf{ doe } x := x+1) = \emptyset$$

Dan geldt ledigerwijs[1] voor alle

$$b' \in T(b, \textbf{zolang } x > 0 \textbf{ doe } x := x+1)$$

dat $M, b' \models x = -99$.

17.3 Axioma's en afleidingsregels

Je hebt nu gezien hoe predikatenlogica kan worden gebruikt om correctheidsbeweringen te geven over programma's. Nu volgt een korte schets van hoe we formeel in de specificatie-logica (ook wel *Floyd–Hoare logica* genoemd naar de uitvinders) kunnen redeneren. De Floyd–Hoare logica voor onze voorbeeldtaal is een axiomatisch systeem met één axioma en een aantal afleidingsregels.

Het axioma heeft betrekking op de toekenningsopdracht, en heet daarom het *toekenningsaxioma*. Het luidt als volgt:

Axioma 17.1 (Toekenningsaxioma)

$$\{[t/v]\varphi\} \; v := t \; \{\varphi\}.$$

Hierbij is v een willekeurige variabele, t een willekeurige term, en φ een willekeurige predikatenlogische formule. De definitie van $[t/v]$ heb je in Opgave 11.13 zelf gegeven. Enkele voorbeelden:

Voorbeeld 17.10 $\vdash \{y = 3\}$ `x := 3` $\{y = x\}$.
Ga na dat deze specificatie een voorbeeld is van een toepassing van het toekenningsaxioma. Elk axioma is een stelling; dit geven we aan met behulp van $\vdash$.

Voorbeeld 17.11 $\vdash \{x - 1 = N\}$ `x := x - 1` $\{x = N\}$.
Wat deze specificatie zegt is: als x na de toekenning een waarde N heeft, dan moet de waarde van $x - 1$ voor de toekenning gelijk zijn geweest aan N. Met andere woorden: als x na de toekenning de waarde N heeft, dan moet de waarde daarvoor $N + 1$ zijn geweest.

Wanneer je het toekenningsaxioma niet erg plausibel vindt (omdat de substitutie 'van achter naar voren' plaatsvindt) sta je niet alleen. Echter, $\{\varphi\}\ v := t\ \{[t/v]\varphi\}$ is *niet* correct, zoals uit het volgende voorbeeld blijkt:

Voorbeeld 17.12 $\vdash \{x = 0\}$ `x := 1` $\{1 = 0\}$
Deze specificatie, die direct volgt uit $\{\varphi\}\ v := t\ \{[t/v]\varphi\}$, is uiteraard onjuist. Het begripsprobleem rond het toekenningsaxioma maakt duidelijk dat de notie *toekennen van een nieuwe waarde aan een variabele* minder simpel is dan op het eerste gezicht lijkt.

De afleidingsregels van Floyd–Hoare logica hebben de volgende vorm:

$$\frac{\begin{array}{l}\vdash S_1\\ \vdots\\ \vdash S_n\end{array}}{\vdash S.}$$

Wat dit zegt is dat $\vdash S$ kan worden afgeleid uit $\vdash S_1$ tot en met $\vdash S_n$. We geven een aantal voorbeelden.

Afleidingsregel 17.1 (Versterking beginconditie)

$$\frac{\begin{array}{ll}\vdash & \varphi \to \psi\\ \vdash & \{\psi\}\ \pi\ \{\chi\}\end{array}}{\vdash \{\varphi\}\ \pi\ \{\chi\}.}$$

In feite staat $\vdash \varphi \to \psi$ hier voor een willekeurige implicatie die een stelling is in een theorie die betrekking heeft op het toepassingsdomein van de programma's (in het geval van **Impertaal**: de theorie van de gehele getallen). Een heel simpel voorbeeld: $\vdash x = N + 1 \to x - 1 = N$. We kunnen de regel bij voorbeeld als volgt gebruiken in een afleiding:

$$\frac{\begin{array}{ll}\vdash & x = N + 1 \to x - 1 = N\\ \vdash & \{x - 1 = N\}\ \mathtt{x := x - 1}\ \{x = N\}\end{array}}{\vdash \{x = N + 1\}\ \mathtt{x := x - 1}\ \{x = N\}.}$$

Hier is een volgende afleidingsregel:

Afleidingsregel 17.2 (Afzwakking eindconditie)

$$\frac{\begin{array}{ll}\vdash & \{\varphi\}\ \pi\ \{\psi\}\\ \vdash & \psi \to \chi\end{array}}{\vdash \{\varphi\}\ \pi\ \{\psi\}.}$$

Voor elke formule ψ hebben we de volgende predikatenlogische stelling: $\vdash \psi \to \top$. Gecombineerd met de zojuist gegeven regel betekent dit dat we voor willekeurige beginvoorwaarde φ en willekeurig programma π af kunnen leiden:

$$\vdash \{\varphi\}\ \pi\ \{\top\}.$$

In het algemeen maken de twee afleidingsregels *versterking van de beginvoorwaarde* en *afzwakking van de eindvoorwaarde* het mogelijk om stellingen over het toepassingsdomein te verweven in correctheidsredeneringen.

De nu volgende regels hebben meer specifiek betrekking op eigenschappen van de **Impertaal** programma's zelf.

Afleidingsregel 17.3 (Opeenvolgingsregel)

$$\frac{\begin{array}{ll}\vdash & \{\varphi\}\ \pi_1\ \{\psi\}\\ \vdash & \{\psi\}\ \pi_2\ \{\chi\}\end{array}}{\vdash \{\varphi\}\ \pi_1;\pi_2\ \{\chi\}.}$$

Deze regel geeft aan wat het effect is van het na elkaar uitvoeren van **Impertaal** opdrachten (of: programma's). Herhaald gebruik van de opeenvolgingsregel produceert correctheidsbeweringen voor **Impertaal** opeenvolgings-opdrachten (daarbij worden `begin` en `einde` respectievelijk links en rechts aan de opdrachtenreeks toegevoegd).

Afleidingsregel 17.4 ('Als–dan' regel)

$$\frac{\begin{array}{ll}\vdash & \{\varphi \wedge \sigma\}\ \pi\ \{\chi\}\\ \vdash & (\varphi \wedge \neg\sigma) \to \chi\end{array}}{\vdash \{\varphi\}\ (\textbf{als}\ \ \sigma\ \textbf{dan}\ \pi)\ \{\chi\}.}$$

Deze regel geeft aan wat het effect is van het uitvoeren van een **als dan** opdracht.

Opgave 17.9 Gebruik de **'Als–dan'** regel om het voorbeeldprogramma (`als x < 0 dan x := -x`) van een specificatie te voorzien.

Afleidingsregel 17.5 ('Als–dan–anders' regel)

$$\frac{\begin{array}{ll}\vdash & \{\varphi \wedge \sigma\}\ \pi_1\ \{\psi\}\\ \vdash & \{\varphi \wedge \neg\sigma\}\ \pi_2\ \{\psi\}\end{array}}{\vdash \{\varphi\}\ (\textbf{als}\ \sigma\ \textbf{dan}\ \pi_1\ \textbf{anders}\ \pi_2)\ \{\psi\}.}$$

Opgave 17.10 Gebruik de **'Als–dan–anders'** regel om het voorbeeldprogramma (`als x > y dan m := x anders m := y`) van een specificatie te voorzien.

Afleidingsregel 17.6 ('Zolang' regel)

$$\frac{\vdash \{\varphi \wedge \sigma\}\ \pi\ \{\varphi\}}{\vdash \{\varphi\}\ \textbf{zolang}\ \sigma\ \textbf{doe}\ \pi\ \{\varphi \wedge \neg\sigma\}.}$$

Als je het stap voor stap toepassen van de Floyd–Hoare axioma's en regels nogal geestdodend vindt dan bewijst dat dat je geen computer bent. Het inzicht dat regeltoepassingen van het slag dat we zojuist hebben gedemonstreerd geknipt zijn voor de computer leidt tot het idee om correctheidsbewijzen voor imperatieve programma's te gaan *mechaniseren*. Het idee is dan een geannoteerd programma in de computer te stoppen en het vervolgens aan de machine over te laten om de condities die met behulp van Floyd–Hoare logica uit de annotaties volgen door te rekenen. De annotaties waarmee we beginnen specificeren wat ons bij het schrijven van het programma voor ogen stond; van die oogmerken kan een computer uiteraard geen weet hebben.

Opgave 17.11 Bewijs de semantische geldigheid van de volgende Hoare tripels.

1. $\{x = 3\}\ \ y := x * x\ \ \{y = 9\}$.
2. $\{x = 3\}\ \ y := z * z\ \ \{y = z^2\}$.
3. $\{x \leq 3\}$ **als** $x \leq 4$ **dan** $y := x$ **anders** $y := 2 * x$ $\{xy \leq 17\}$.
4. $\{x \leq 3\}$ **als** $x \leq 1$ **dan** $y := x$ **anders** $y := x - 1$ $\{xy \leq 9\}$.

Stelling 17.1 (Gezondheid) *De vijf afleidingsregels in de Hoare calculus zijn gezond.*

Gezond wil zeggen: semantisch correct. De bewijsregels voeren dus ware Hoare tripels over in andere ware Hoare tripels. We bewijzen de stelling voor de assignment-statement en de while statement. De overige gevallen laten we ter oefening. (De oplossingen staan achterin.)

Voor de assignment-statement moeten we bewijzen dat voor elke formule φ, term t en variable x het tripel

$$\{[t/x]\varphi\}\ \ x := t\ \ \{\varphi\}$$

geldig is. Laat daarvoor $\mathscr{M}$ en b willekeurig, en stel $\mathscr{M}, b \models [t/x]\varphi$. We moet bewijzen dat, als $b' \in T(b, x := t)$, dan $\mathscr{M}, b' \models \varphi$. Volgens Definitie 17.2 geldt

$$T(b, x := t) = \{b(x|W_{M,b}(t))\}$$

dus moeten we bewijzen dat $\mathscr{M}, b(x|W_{M,b}(t)) \models \varphi$. Welnu, dát volgt precies uit het substitutielemma! (blz. 137):

$$M, b \models [t/x]\varphi \quad \Leftrightarrow \quad \mathscr{M}, b(x|W_{M,b}(t)) \models \varphi.$$

Voor de while-statement moeten we bewijzen dat, als

$$\{\varphi \wedge \sigma\}\ \pi\ \{\varphi\}, \tag{15}$$

geldt, dat dan ook

$$\{\varphi\}\ \textbf{zolang}\ \sigma\ \textbf{doe}\ \pi\{\varphi \wedge \neg\sigma\}. \tag{16}$$

geldt. Stel dus dat (15) geldt. Dat betekent dat voor alle $\mathscr{M}, b$, als

$$\mathscr{M}, b \models \varphi \wedge \sigma \quad \text{en} \quad b' \in T(b, \pi) \tag{17}$$

dan ook

$$\mathscr{M}, b' \models \varphi. \tag{18}$$

Stel, om (16) te bewijzen, dat

$$\mathscr{M}, b \models \varphi, \tag{19}$$

en stel dat

$$b' \in T(b, \textbf{zolang}\ \sigma\ \textbf{doe}\ \pi). \tag{20}$$

Te bewijzen

$$M, b' \models \varphi \wedge \neg\sigma. \tag{21}$$

Volgens Definitie 17.2 geldt voor (20) dat er een eindig rijtje toestanden $b_1, \ldots, b_n$ bestaat, zó dat $b_1 = b$, $b_n = b'$, en

$$\begin{array}{rcl}
M, b_1 \models \sigma & \text{en} & b_1 = b \\
M, b_2 \models \sigma & \text{en} & b_2 \in T(b_1, \pi) \\
M, b_3 \models \sigma & \text{en} & b_3 \in T(b_2, \pi) \\
\vdots & \vdots & \vdots \\
M, b_{n-1} \models \sigma & \text{en} & b_{n-1} \in T(b_{n-2}, \pi) \\
M, b_n \not\models \sigma & \text{en} & b_n \in T(b_{n-1}, \pi)\ \text{en}\ b_n = b'
\end{array}$$

Uit (19) volgt dan $M, b_1 \models \varphi \wedge \sigma$, en met (15), oftewel (17) en (18), volgt dan $M, b_2 \models \varphi \wedge \sigma$. Dit voortzettend (formeel: met volledige inductie) krijgen we

$$\begin{array}{rcl}
M, b_1 \models \varphi \wedge \sigma & \text{en} & b_1 = b \\
M, b_2 \models \varphi \wedge \sigma & \text{en} & b_2 \in T(b_1, \pi) \\
M, b_3 \models \varphi \wedge \sigma & \text{en} & b_3 \in T(b_2, \pi) \\
\vdots & \vdots & \vdots \\
M, b_{n-1} \models \varphi \wedge \sigma & \text{en} & b_{n-1} \in T(b_{n-2}, \pi) \\
M, b_n \models \varphi & \text{en} & \\
M, b_n \not\models \sigma & \text{en} & b_n \in T(b_{n-1}, \pi)\ \text{en}\ b_n = b'
\end{array}$$

Het laatste geeft natuurlijk

$$M, b_n \models \varphi \wedge \neg\sigma$$

aangezien $b_n = b'$ en we (21) moesten bewijzen, zijn we klaar. □

Opgave 17.12 Bewijs de gezondheid, i.e., semantische correctheid, van de volgende regels.

1. Opeenvolgingsregel.
2. Als-dan regel.
3. Als-dan-anders regel.

17.4 Bewijzen

De vraag nu is hoe de correctheid van wat ingewikkeldere programma's bewezen kan worden. Laten we met concatenatie beginnen.

Concatenatie

Omdat de assignment-regel wezenlijk van achter naar voren werkt, is het 't gemakkelijkst om bij een opeenvolging van statements de post-conditie van een reeks assignments als het ware van achter naar voren "door het programma heen te trekken".

Voorbeeld 17.13 Bewijs de correctheid van

$\{w \le 5, z = 3\}$
$x := 2;\ y = x + z;\ u := 4;$
$x = u - w + x + z;\ w = x + y;$
$\{w \ge 8\}$

Als je het voorbeeld eerst even "op z'n janboerenfluitjes" narekent, dan krijg je achtereenvolgens $x = 2$, $y = 5$, $u = 4$,

$$x = 4 - w + 2 + 3 \ge 4,$$

en $w \ge 9$. Aangezien $w \ge 9 \rightarrow w \ge 8$, klopt de bewering. (Had je al door dat "op z'n janboerenfluitjes" eigenlijk volgens Def. 17.2 verloopt?)

Nu formeel. Eerst schrijven we het programma zó op dat er correctheid-condities tussen kunnen worden geplaatst:

$\{w \le 5, z = 3\}$	pre-conditie
$x := 2;$	
$y = x + z;$	
$u := 4;$	
$x = u - w + x + z;$	
$w = x + y;$	
$\{w \ge 8\}$	post-conditie

Het alleen maar opschrijven van een tripel betekend nog niet dat het *correct* is, dat willen we nu juist bewijzen! Een correctheids-bewijs leveren we door de post-conditie overeenkomstig de assignment-regel stap voor stap *naar boven* te werken. Eerst wordt de assignment-regel toegepast op de laatste statement $w = x + y$ en de laatste conditie $\{w \ge 8\}$:

$\{w \le 5, z = 3\}$	pre-conditie
$x := 2;$	
$y = x + z$	
$u := 4$	
$x = u - w + x + z$	
$\{x + y \ge 8\}$	assignment-regel
$w = x + y$	
$\{w \ge 8\}$	post-conditie

Vervolgens wordt de assignment-regel toegepast op de statement $x = u - w + x + z$ en de zojuist gegenereerde conditie $\{x + y \ge 8\}$:

$\{w \le 5, z = 3\}$	pre-conditie
$x := 2;$	
$y = x + z$	
$u := 4$	
$\{u - w + x + z + y \ge 8\}$	assignment-regel
$x = u - w + x + z$	
$\{x + y \ge 8\}$	assignment-regel
$w = x + y$	
$\{w \ge 8\}$	post-conditie

Op dezelfde manier werken we steeds verder naar boven. Op enig moment is het toegestaan condities algebraïsch te vereenvoudigen.

$\{w \le 5, z = 3\}$	pre-conditie
$\{4 - w + 2 \cdot 2 + 2z \ge 8\}$	
$x := 2;$	
$\{4 - w + 2x + 2z \ge 8\}$	vereenvoudiging
$\{4 - w + x + z + x + z \ge 8\}$	assignment-regel
$y = x + z$	
$\{4 - w + x + z + y \ge 8\}$	assignment-regel
$u := 4$	
$\{u - w + x + z + y \ge 8\}$	assignment-regel
$x = u - w + x + z$	
$\{x + y \ge 8\}$	assignment-regel
$w = x + y$	
$\{w \ge 8\}$	post-conditie

Dat mag, omdat we op dit moment bezig zijn een correctheidsbewijs te leveren. In zo'n situatie mag elke (algebraïsche) vereenvoudiging zonder logische verantwoording worden opgeschreven (anders dan: "vereenvoudiging"). In het hoofdstuk over predikaatlogica werden we nog geacht deze stappen als oefening formeel te bewijzen. In deze context hoeft dat dus niet meer.
We gaan verder:

$\{w \le 5, z = 3\}$	pre-conditie
$\{4 - w + 2 \cdot 2 + 2z \ge 8\}$	assignment-regel
$x := 2;$	
$\{4 - w + 2x + 2z \ge 8\}$	vereenvoudiging
$\{4 - w + x + z + x + z \ge 8\}$	assignment-regel
$y = x + z$	
$\{4 - w + x + z + y \ge 8\}$	assignment-regel
$u := 4$	
$\{u - w + x + z + y \ge 8\}$	assignment-regel
$x = u - w + x + z$	
$\{x + y \ge 8\}$	assignment-regel
$w = x + y$	
$\{w \ge 8\}$	post-conditie

Na $[2/x]2x$ te hebben vereenvoudigd

1. $\{w \le 5, z = 3\}$	versterking, 2
2. $\{w \ge 2z\}$	vereenvoudiging, 3
3. $\{4 - w + 2 \cdot 2 + 2z \ge 8\}$	assignment-regel 4, 5
4. $x := 2;$	
5. $\{4 - w + 2x + 2z \ge 8\}$	vereenvoudiging, 6
6. $\{4 - w + x + z + x + z \ge 8\}$	assignment-regel 7, 8
7. $y = x + z$	
8. $\{4 - w + x + z + y \ge 8\}$	assignment-regel 9, 10
9. $u := 4$	
10. $\{u - w + x + z + y \ge 8\}$	assignment-regel 11,12
11. $x = u - w + x + z$	
12. $\{x + y \ge 8\}$	assignment-regel 13,14
13. $w = x + y$	
14. $\{w \ge 8\}$	post-conditie

moet als het goed is de omhooggewerkte post-conditie $\{w \geq 2z\}$ geïmpliceerd worden door de pre-conditie $\{w \leq 5, z = 3\}$. Dat is inderdaad het geval, zodat we, na regelnummering, meteen een formeel correctheids-bewijs hebben.

Merk op dat de rechtvaardigingen in het verkregen correctheidsbewijs nog steeds van achter naar voren lopen. Voor rijen assignments-statements is dat in het algemeen ook zo. Voor if-then-else en while-statements zijn rechtvaardigingen niet meer lineair. Dit wegvallen van lineariteit wordt veroorzaakt door de opbouw van de toegepaste regels.

Opgave 17.13 Geef correctheidsbewijzen van de volgende Hoare tripels:

1. $\{x = 2, y = 3\}\ x := 2;\ y := 8;\ x := 5;\ y := 7\ \{x + y = 12\}$
2. $\{x = 2, y = 3\}\ x := x + y;\ x := x - y;\ y := x - y\ \{y = -1\}$
3. $\{x = 2, y = 3\}\ x := x + y;\ x := x - y;\ y := x - y\ \{xy = -2\}$
4. $\{x = 2, w = 3\}\ \ w := x - w;\ x = -2 * w\ \{x = 2\}$
5. $\{x = 2, w \leq 3\}\ \ w := x - w;\ x = -2 * w\ \{x \leq 3\}$

Als-dan-anders

Vervolgens willen we programma's met if-then-else constructies correct kunnen bewijzen. Met de huidige versie van de if-then-else regel gaat dat zeer moeizaam, omdat bij deze regel de pre-condities van een bepaalde vorm moeten zijn, i.e., van de vorm $\varphi \wedge \sigma$ en $\varphi \wedge \neg\sigma$. Willekeurig omhoog geduwde condities (φ_1 resp. φ_2) bezitten meestal niet deze vorm. Om niet meer gebonden te zijn aan een vast formaat pre-condities gaan we een alternatieve if-then-else regel opstellen.

Stelling 17.2 *De als-dan-anders regel is gelijkwaardig aan de volgende alternatieve als-dan-anders regel*

$$\frac{\vdash\ \{\varphi_1\}\ \pi_1\ \{\psi\} \qquad \vdash\ \{\varphi_2\}\ \pi_2\ \{\psi\}}{\vdash\ \{(\sigma \to \varphi_1) \wedge (\neg\sigma \to \varphi_2)\}\ \textbf{als}\ \sigma\ \textbf{dan}\ \pi_1\ \textbf{anders}\ \pi_2\ \{\psi\}.}$$

Merk op dat de pre-condities $\{\varphi_1\}$ en $\{\varphi_2\}$ van π_1 resp. π_2 geen logische structuur bezitten, en er dus in principe alles voor kan worden ingevuld.

Bewijs: We gaan eerst de nieuwe regel uit de oude regel afleiden. Stel dus dat $\{\varphi_1\}\ \pi_1\ \{\psi\}$ en $\{\varphi_2\}\ \pi_2\ \{\psi\}$. Nu lijken we op het eerste gezicht niet verder te kunnen omdat de pre-condities van de nieuwe regel, te weten φ_1 en φ_2, niet in de vorm van de pre-condities van de oorspronkelijke regel, $\varphi \wedge \sigma$ resp. $\varphi \wedge \neg\sigma$ staan, en de oorspronkelijke regel dus niet direct kan worden toegepast. We moeten dus een formule χ vinden, zó dat

$$\varphi_1 \equiv \chi \wedge \sigma \quad \text{en} \quad \varphi_1 \equiv \chi \wedge \neg\sigma, \tag{22}$$

zodat we de oorspronkelijke regel wél kunnen toepassen, en ook nog geldt dat

$$\chi = (\sigma \to \varphi_1) \wedge (\neg\sigma \to \varphi_2) \tag{23}$$

als preconditie van de conclusie van de gevolgtrekking optreedt. Dit zijn nogal wat eisen, maar we zullen zien dat ze makkelijk kunnen worden ingewilligd. Immers, (23) is niet alleen een eis maar, omdat χ geïsoleerd aan de linkerkant staat, ook meteen een definitie. Het blijkt dat de χ die voldoet aan (23) gelukkig ook voldoet aan (22), zodat de oorspronkelijke regel meteen kan worden toegepast om de gewenste gevolgtrekking te krijgen. (Is het niet meteen duidelijk dan kun je desnoods de simulatie van de nieuwe regel middels de oude regel zelf "naspelen".)

Stel, omgekeerd, dat de nieuwe regel geldt. We proberen nu de oorspronkelijke regel af te leiden. Stel dus dat $\{\varphi \wedge \sigma\}\ \pi_1\ \{\psi\}$ en $\{\varphi \wedge \neg\sigma\}\ \pi_2\ \{\psi\}$. Volgens de nieuwe regel geldt dan

$$\{(\sigma \to (\varphi \wedge \sigma)) \wedge (\neg\sigma \to (\varphi \wedge \neg\sigma))\}\ \textbf{als}\ \sigma\ \textbf{dan}\ \pi_1\ \textbf{anders}\ \pi_2\ \{\psi\}.$$

Omdat

$$(\sigma \to (\varphi \wedge \sigma)) \wedge (\neg\sigma \to (\varphi \wedge \neg\sigma)) \equiv \varphi$$

volgt de oorspronkelijke regel. □

We geven nu een voorbeeld hoe de nieuwe if-then-else regel kan worden toegepast:

Voorbeeld 17.14 Bewijs de correctheid van

$\{x \leq 10\}$
als $x \leq 5$ **dan** $y = x$ **anders** $y = -x$;
$\{|y| = |x|\}$

Eerst schrijven we het programma zó op dat er correctheid-condities tussen kunnen worden geplaatst:

$\{x \leq 10\}$
als $x \leq 5$
  dan
  $y = x$
  anders
  $y = -x$
  ;
$\{|y| = |x|\}$

Zie nu hoe vervolgens de post-conditie $\{|y| = |x|\}$ naar boven kan worden gebracht:

$\{x \leq 10\}$
als $x \leq 5$
  dan
  $y = x$
  $\{|y| = |x|\}$
  anders
  $y = -x$
  $\{|y| = |x|\}$
  ;
$\{|y| = |x|\}$

Merk op dat in één stap de conditie $\{|y| = |x|\}$ zich verdeeld over beide if-then-else takken.

$\{x \le 10\}$
als $x \le 5$
  dan
  $\{|x| = |x|\}$
  $y = x$
  $\{|y| = |x|\}$
  anders
  $\{|-x| = |x|\}$
  $y = -x$
  $\{|y| = |x|\}$
  ;
$\{|y| = |x|\}$

Nu kunnen de twee condities $\{|x| = |x|\}$ en $\{|-x| = |x|\}$ met behulp van de nieuwe regel samen worden genomen middels de vorm $(\sigma \to \varphi_1) \wedge (\neg\sigma \to \varphi_2)$, waarbij

$$\sigma = x \le 5$$
$$\varphi_1 = |x| = |x|$$
$$\varphi_2 = |-x| = |x|$$

Overeenkomstig:

$\{x \le 10\}$
$\{\top\}$
$\{(x \le 5 \to |x| = |x|) \wedge (x \not\le 5 \to |-x| = |x|)\}$
als $x \le 5$
  dan
  $\{|x| = |x|\}$
  $y = x$
  $\{|y| = |x|\}$
  anders
  $\{|-x| = |x|\}$
  $y = -x$
  $\{|y| = |x|\}$
  ;
$\{|y| = |x|\}$

De voorlaatste conditie is een tautologie en hebben we dus meteen maar vertaald in $\top$. (De oorspronkelijke pre-conditie is dus eigenlijk te sterk.) Omdat $x \le 10 \Rightarrow \top$, mogen we concluderen dat we uit deze lijst van programma-instructies en condities een bewijs kunnen maken:

1.	$\{x \le 10\}$	versterking, 2
2.	$\{\top\}$	vereenvoudiging, 3
3.	$\{(x \le 5 \to \|x\| = \|x\|) \wedge \ldots$	als-dan-anders regel′, 6-8, 10-12
4.	**als** $x \le 5$	
5.	**dan**	
6.	$\{\|x\| = \|x\|\}$	assignment-regel, 7,8
7.	$y = x$	
8.	$\{\|y\| = \|x\|\}$	postconditie, 14
9.	**anders**	
10.	$\{\|-x\| = \|x\|\}$	assignment-regel, 11,12
11.	$y = -x$	
12.	$\{\|y\| = \|x\|\}$	postconditie, 14
13.	;	
14.	$\{\|y\| = \|x\|\}$	als-dan-anders regel′, 6-8, 10-12

Merk op dat de verantwoordingen niet meer lineair lopen. Dit wordt veroorzaakt door de regelopbouw van de als-dan-anders regel′.

De verantwoording "als-dan-anders regel′" slaat op de alternatieve als-dan-anders regel, die staat uitgelegd op blz. 208.

Opgave 17.14 Geef correctheidsbewijzen van de volgende Hoare tripels:

1. $\{x = 2, y = 3\}$
   als $(x = 2)$ **dan** $x := y$ **anders** $y := x$; $\{x = 3\}$
2. $\{x = 2, y = 3\}$
   als $(x \ne y)$ **dan** $x := 1/(y - x)$ **anders** $y := x$; $\{x = 1\}$
3. $\{x = 2, y = 3\}$
   als $(x = y)$ **dan** $x := 1/(y - x)$ **anders** $y := x$; $\{x = 1\}$
4. $\emptyset$ **als** $(x \ge 0)$ **dan** $y = x$ **anders** $y := -x$; $\{y = |x|\}$
5. $\emptyset$ **als** $(x \ge y)$ **dan** $m := x$ **anders** $m := y$; $\{m = \max\{x, y\}\}$

Zolang

Vervolgens willen we programma's met while-constructies correct kunnen bewijzen. Dit is verreweg het meest lastige geval, omdat de 'Zolang'-regel niet, zoals de als-dan-anders regel, in een schema kan worden gegoten waarbij pre- en post-condities een vrij formaat mogen hebben. Dus als we de algemene vorm van

$$\{\varphi\}\ \textbf{zolang}\ \sigma\ \textbf{doe}\ \pi\ \{\psi\}$$

willen bewijzen, dan zit er helaas niets anders op dan zelf een invariant η te verzinnen, zó dat

$$\varphi \to \eta,\ \ \{\eta\}\ \textbf{zolang}\ \sigma\ \textbf{doe}\ \pi\ \{\eta \wedge \neg\sigma\},\ \ (\eta \wedge \neg\sigma) \to \psi$$

Het verzinnen van zo'n invariant is een *kunst:* η moet zwak genoeg zijn om te worden geïmpliceerd door de pre-conditie φ, maar sterk genoeg zijn om samen met $\neg\sigma$ de post-conditie ψ te impliceren.

Voorbeeld 17.15 Bewijs de correctheid van

$\{x = 0\}$
zolang $x < 5$ **doe**
  $\{\ x := x + 50;\ x := x - 49;\ \}$
$\{x = 5\}$

We schrijven eerst

$\{x = 0\}$
zolang $x < 5$ **doe** {
  $x := x + 50;$
  $x := x - 49;$
}
$\{x = 5\}$

Wat zou aan aan het begin en eind van elke slag kunnen gelden, i.e., wat voor invariant η zouden we kunnen kiezen? We kunnen in ieder geval beweren dat na elke slag x altijd kleiner of gelijk blijft aan 5. *Merk op dat in de while-body zélf best mag gelden dat $x \not\leq 5$!* Laten we dus de invariant

$$\eta = x \leq 5$$

proberen. We moet nu in ieder geval eerst controleren of $\eta \wedge \neg\sigma$ sterk genoeg is om de post-conditie te impliceren. Dat is het geval:

$$\begin{aligned} & (\eta \wedge \neg\sigma) \rightarrow x = 5 \\ \Leftrightarrow \quad & (x \leq 5 \wedge \neg(x < 5)) \rightarrow x = 5. \end{aligned}$$

Vervolgens moeten we inschatten of η zwak genoeg is om door $x = 0$ te worden geïmpliceerd. (Inschatten, omdat we bij lange programma's niet precies kunnen voorspellen welke conditie er precies voor de while-statement terecht gaat komen. Hier kunnen we dat makkelijk zeggen: $x = 0$.) Dat is ook het geval:

$$\begin{aligned} & x = 0 \rightarrow \eta \\ \Leftrightarrow \quad & x = 0 \rightarrow x \leq 5. \end{aligned}$$

We gaan nu de post-conditie omhoog halen:

$\{x = 0\}$
zolang $x < 5$ **doe** {
  $x := x + 50;$
  $x := x - 49;$
}
$\{x \leq 5 \wedge \neg(x < 5)\}$
$\{x = 5\}$

Vervolgens

$\{x = 0\}$
zolang $x < 5$ **doe** {
  $x := x + 50;$
  $x := x - 49;$
  $\{x \leq 5\}$
}
$\{x \leq 5 \wedge \neg(x < 5)\}$
$\{x = 5\}$

Invariant $\{x \leq 5\}$ door de body van de while-statement heen halen en vereenvoudigen levert:

$\{x = 0\}$
zolang $x < 5$ **doe** {
  $\{x + 1 \leq 5\}$
  $\{x + 50 - 49 \leq 5\}$
  $x := x + 50;$
  $\{x - 49 \leq 5\}$
  $x := x - 49;$
  $\{x \leq 5\}$
}
$\{x \leq 5 \wedge \neg(x < 5)\}$
$\{x = 5\}$

De bovenste conditie $\{x + 1 \leq 5\}$ schrijven in de vorm $\eta \wedge \sigma$ levert:

$\{x = 0\}$
zolang $x < 5$ **doe** {
  $\{x \leq 5 \wedge x < 5\}$
  $\{x + 1 \leq 5\}$
  $\{x + 50 - 49 \leq 5\}$
  $x := x + 50;$
  $\{x - 49 \leq 5\}$
  $x := x - 49;$
  $\{x \leq 5\}$
}
$\{x \leq 5 \wedge \neg(x < 5)\}$
$\{x = 5\}$

Volgens de 'Zolang'-regel mag de invariant $\eta = x \leq 5$ verder naar boven worden gehaald:

$\{x = 0\}$
$\{x \leq 5\}$
zolang $x < 5$ **doe** {
  $\{x \leq 5 \wedge x < 5\}$
  $\{x + 1 \leq 5\}$
  $\{x + 50 - 49 \leq 5\}$
  $x := x + 50;$
  $\{x - 49 \leq 5\}$
  $x := x - 49;$
  $\{x \leq 5\}$
}
$\{x \leq 5 \wedge \neg(x < 5)\}$
$\{x = 5\}$

We zijn nu klaar. Omdat de naar boven gehaalde post-conditie geïmpliceerd word door de pre-conditie, kunnen we deze lijst omzetten in een bewijs:

1.	$\{x = 0\}$	versterking, 2
2.	$\{x \leq 5\}$	Zolang-regel, 4-10
3.	**zolang** $x < 5$ **doe** {	
4.	$\{x \leq 5 \wedge x < 5\}$	invariant en guard
5.	$\{x + 1 \leq 5\}$	vereenvoudiging, 6
6.	$\{x + 50 - 49 \leq 5\}$	assignment, 7,8
7.	$x := x + 50;$	
8.	$\{x - 49 \leq 5\}$	assignment, 9,10
9.	$x := x - 49;$	
10.	$\{x \leq 5\}$	invariant
11.	}	
12.	$\{x \leq 5 \wedge \neg(x < 5)\}$	Zolang-regel, 4-10
13.	$\{x = 5\}$	verzwakking, 12

Voorbeeld 17.16 Bewijs de correctheid van

$\{\top\}$
$y := 1;$
$z := 0;$
zolang z <> x **doe** {
  $z := z + 1;$
  $y := y * z;$
}
$\{y = x!\}$

Om een geschikte invariant te vinden bekijken we het verloop van een bedeling b gedurende een run als bij aanvang $b(x) = 5$:

	$b(x)$	$b(y)$	$b(z)$	$z <> x$
	5	–	–	true
	5	1	–	true
ingang 'Zolang'	5	1	0	true
	5	1	1	true
	5	2	2	true
	5	6	3	true
	5	24	4	true
uitgang 'Zolang'	5	120	5	false

(24)

Wat verandert niet tussen "ingang" en "uitgang"? Uit de tabel zien we na enig staren dat $y = z!$ altijd geldt. Neem dus als invariant

$$\eta = (y = z!).$$

Is η nu sterk genoeg om samen met $\neg\sigma$ de post-conditie te impliceren? Ja:

$$\begin{aligned} & (\eta \wedge \neg\sigma) \rightarrow (y = x!) \\ \Leftrightarrow \quad & (y = z! \wedge \neg(x <> z)) \rightarrow (y = x!) \\ \Leftrightarrow \quad & (y = z! \wedge x = z) \rightarrow (y = x!). \end{aligned}$$

Vervolgens moeten we inschatten of η zwak genoeg is om door de conditie vlak boven de while-statement (een conditie die we op dit moment officieel nog niet weten) te worden geïmpliceerd. Dat is ook het geval:

$$\begin{aligned} & (y = 1 \wedge z = 0) \rightarrow \eta \\ \Leftrightarrow \quad & (y = 1 \wedge z = 0) \rightarrow (y = z!). \end{aligned}$$

We gaan nu de post-conditie omhoog halen. Eerst schrijven we de $\eta \wedge \neg\sigma$ achter de while-loop:

$\{\top\}$
$y := 1;$
$z := 0;$
zolang $z <> x$ **doe** {
  $z := z + 1;$
  $y := y * z;$
}
$\{y = z! \wedge \neg(x <> z)\}$
$\{y = x!\}$

Vervolgens plaatsen we de invariant η zelf achter de body van de while-loop:

$\{\top\}$
$y := 1;$
$z := 0;$
zolang $z <> x$ **doe** {
  $z := z + 1;$
  $y := y * z;$
  $\{y = z!\}$
}
$\{y = z! \wedge \neg(x <> z)\}$
$\{y = x!\}$

De invariant $\{y = z!\}$ door de body van de while-statement heen halen en vereenvoudigen levert:

$\{\top\}$
$y := 1;$
$z := 0;$
zolang $z <> x$ **doe** {
  $z := z + 1;$
  $\{y * z = z!\}$
  $y := y * z;$
  $\{y = z!\}$
}
$\{y = z! \wedge \neg(x <> z)\}$
$\{y = x!\}$

Vervolgens

$\{\top\}$
$y := 1;$
$z := 0;$
zolang $z <> x$ **doe** {
  $\{y * (z + 1) = (z + 1)!\}$
  $z := z + 1;$
  $\{y * z = z!\}$
  $y := y * z;$
  $\{y = z!\}$
}
$\{y = z! \wedge \neg(x <> z)\}$
$\{y = x!\}$

De bovenste conditie $\{y * (z + 1) = (z + 1)!\}$ in de vorm $\eta \wedge \sigma$ schrijven levert:

$\{\top\}$
$y := 1;$
$z := 0;$
zolang $z <> x$ **doe** {
  $\{(y = z!) \wedge (z \neq x)\}$
  $\{y * (z + 1) = (z + 1)!\}$
  $z := z + 1;$
  $\{y * z = z!\}$
  $y := y * z;$
  $\{y = z!\}$
}
$\{y = z! \wedge \neg(x <> z)\}$
$\{y = x!\}$

Inderdaad geldt

$$(y = z!) \wedge (z \neq x) \Rightarrow y * (z + 1) = (z + 1)!$$

Volgens de 'Zolang'-regel mogen we de invariant $\eta = (y = z!)$ verder naar boven halen:

$\{\top\}$
$y := 1;$
$z := 0;$
$\{y = z!\}$
zolang $z <> x$ **doe** {
  $\{(y = z!) \wedge (z \neq x)\}$
  $\{y * (z + 1) = (z + 1)!\}$

$z := z + 1;$
$\{y * z = z!\}$
$y := y * z;$
$\{y = z!\}$
}
$\{y = z! \wedge \neg(x <> z)\}$
$\{y = x!\}$

Vervolgens enkele keren de assignment regel toepassen levert

$\{\top\}$
$\{1 = 0!\}$
$y := 1;$
$\{y = 0!\}$
$z := 0;$
$\{y = z!\}$
zolang $z <> x$ **doe** {
$\{(y = z!) \wedge (z \neq x)\}$
$\{y * (z + 1) = (z + 1)!\}$
$z := z + 1;$
$\{y * z = z!\}$
$y := y * z;$
$\{y = z!\}$
}
$\{y = z! \wedge \neg(x <> z)\}$
$\{y = x!\}$

Het uiteindelijke correctheidsbewijs ziet er dan zo uit:

1.	$\{\top\}$	vereenvoudiging, 2
2.	$\{1 = 0!\}$	assignment-regel, 3,4
3.	$y := 1;$	
4.	$\{y = 0!\}$	assignment-regel, 5,6
5.	$z := 0;$	
6.	$\{y = z!\}$	Zolang-regel, 8-13
7.	**zolang** $z <> x$ **doe** {	
8.	$\{(y = z!) \wedge (z \neq x)\}$	invariant en guard
9.	$\{y * (z + 1) = (z + 1)!\}$	assignment-regel, 10,11
10.	$z := z + 1;$	
11.	$\{y * z = z!\}$	assignment-regel, 12,13
12.	$y := y * z;$	
13.	$\{y = z!\}$	invariant
14.	}	
15.	$\{y = z! \wedge \neg(x <> z)\}$	Zolang-regel, 8-13
16.	$\{y = x!\}$	post-conditie

Opgave 17.15 Bewijs de volgende Hoare tripels correct. Doe dit als volgt. Stel (voor willekeurige maar niet te grote startwaarden) een executietabel op. (Een voorbeeld is te zien in (24) op blz. 211.) Bekijk vervolgens het verloop van de waarden *binnen* de while loop, en stel op basis van dit verloop een invariant op. Gebruik tenslotte de **zolang** regel (blz. 205) om een bewijs te construeren.

1. Copy1:

$\{x \geq 0\}$
$a := x;\ y := 0;$
zolang $a <> 0$ **doe** {
$y := y + 1;\ a := a - 1;$
}
$\{x = y\}$

2. Copy2

$\{x \geq 0\}$
$y := 0;$
zolang $y <> x$ **doe** {
$y := y + 1;$
}
$\{x = y\}$

3. Multi1

$\{y \geq 0\}$
$a := 0;\ z := 0;$
zolang $a <> y$ **doe** {
$z := z + x;\ a := a + 1;$
}
$\{z = xy\}$

4. Multi2

$\{y = y_0 \wedge y \geq 0\}$
$z = 0;$
zolang $y <> 0$ **doe** {
$z = z + x;\ y = y - 1;$
}
$\{z = xy_0\}$

Opgave 17.16 Wanneer $x, d \in \mathbb{N}$ en $d \neq 0$, kan worden bewezen dat er unieke gehele getallen $q, r \in \mathbb{N}$ zijn, zo dat

$$x = qd + r \text{ en } 0 \leq r < d.$$

Het getal q heet *quotiënt* en het getal r wordt *rest* genoemd. Men schrijft voor de rest r wel:

$$r = x \pmod{d}.$$

– 13 gedeeld door 10 levert 1 als quotiënt en 3 als rest op, omdat $13 = 1 \cdot 10 + 3$.

– 56 gedeeld door 7, levert 8 als quotiënt en 0 als rest op, omdat $56 = 7 \cdot 8 + 0$.

Het volgende programma wordt verondersteld deling met rest uit te voeren. Bewijs de correctheid van dit programma.

$\{y \neq 0\}$
$r := x;\ d := 0;$
zolang $r > y$ **doe** {
$r := r - y;\ d := d + 1;$
}
$\{(x = d \cdot y + r) \wedge (r < y)\}$

17.5 Terminatie

Bovenstaande correctheidsbewijzen behelzen partiële correctheid, dat wil zeggen dat bovenstaande bewijzen laten zien dat, *als* een programma π eindigt, ze dat correct doet. Als we ook nog kunnen bewijzen dat π inderdaad eindigt, dan hebben we een bewijs van volledige correctheid.

Definitie 17.4 (Volledige correctheid)

volledige correctheid = partiële correctheid + terminatie

Hoe bewijzen we terminatie? Als volgt. Om te laten zien dat

$$\textbf{zolang } \sigma \textbf{ doe } \pi$$

ooit zal stoppen, wordt een expressie t gezocht met de volgende eigenschappen:

1. t is geheeltallig.
2. t *daalt* bij elke uitvoering van π.
3. Binnen de while loop geldt $t \geq 0$.

Deze expressie, t, wordt om begrijpelijke redenen een *variant* genoemd. Een geschikt gekozen variant garandeert natuurlijk dat de bijbehorende while-loop ooit zal stoppen. (Ga na!)

De eis dat t daalt dankzij π wordt logisch weergegeven door

$$\{t = x\} \;\; \pi \;\; \{t < x\},$$

waarbij x (om problemen te voorkomen) niet voorkomt in t en π. De eis dat t *binnen de while loop* niet negatief kan worden, wordt logisch weergegeven door

$$\{\sigma \wedge t \geq 0\} \;\; \pi \;\; \{t \geq 0\}.$$

Meestal kan

$$\{\sigma \wedge t \geq 0 \wedge t = x\} \;\; \pi \;\; \{t \geq 0 \wedge t < x\}$$

niet zonder meer worden bewezen, en is de invariant φ van het partiële correctheidsbewijs

$$\{\varphi\} \;\; \textbf{zolang } \sigma \textbf{ doe} \;\; \pi \;\; \{\varphi \wedge \neg\sigma\}$$

nodig. Op deze manier verandert de oorspronkelijke voorwaarde voor partiële correctheid

$$\{\varphi \wedge \sigma\} \;\; \pi \;\; \{\varphi\}$$

in

$$\{\varphi \wedge \sigma \wedge t \geq 0 \wedge t = x\} \;\; \pi \;\; \{\varphi \wedge t \geq 0 \wedge t < x\}. \qquad (25)$$

Dus als (25) bewezen is, dan geeft dat volledige correctheid.

Samenvattend: het tripel

$$\{\varphi \wedge \sigma\} \;\; \pi \;\; \{\varphi\}$$

geeft partiële correctheid, het tripel

$$\{\sigma \wedge t \geq 0 \wedge t = x\} \;\; \pi \;\; \{t \geq 0 \wedge t < x\}.$$

geeft terminatie (maar niet noodzakelijk partiële correctheid), en een combinatie daarvan, (25), geeft volledige correctheid.

Voorbeeld 17.17 Bewijs de volledige correctheid van het programma in Voorbeeld 17.16 op blz. 210.

We hadden al een partieel correctheids-bewijs:

1.	$\{\top\}$	vereenvoudiging, 2
2.	$\{1 = 0!\}$	assignment-regel, 3,4
3.	$y := 1;$	
4.	$\{y = 0!\}$	assignment-regel, 5,6
5.	$z := 0;$	
6.	$\{y = z!\}$	Zolang-regel, 8-13
7.	**zolang** $z <> x$ **doe** {	
8.	$\{(y = z!) \wedge (z \neq x)\}$	invariant en guard
9.	$\{y * (z+1) = (z+1)!\}$	assignment-regel, 10,11
10.	$z := z + 1;$	
11.	$\{y * z = z!\}$	assignment-regel, 12,13
12.	$y := y * z;$	
13.	$\{y = z!\}$	invariant
14.	}	
15.	$\{y = z! \wedge \neg(x <> z)\}$	Zolang-regel, 8-13
16.	$\{y = x!\}$	post-conditie

Wat kunnen we nemen als variant? Dat wil zeggen, wat kunnen we nemen als niet-negatieve dalende expressie t? We zien dat z elke slag stijgt, dus een eerste poging voor de variant is $t = -z$. Echter, we moeten ook nog garanderen dat $t \geq 0$. Daarvoor moeten we beter kijken naar het verloop van z. We zien dat z in het begin gelijk is aan 0 en daarna elke slag 1 stijgt totdat $z = x$. Al die tijd blijft $z \leq x$, dus $x - z \geq 0$. Als variant kunnen dus nemen

$$t = x - z.$$

We proberen dus correct te bewijzen

$\{\top\}$
$y := 1;$
$z := 0;$
zolang $z <> x$ **doe** {
  $\{(y = z!) \wedge (z \neq x) \wedge (x - z \geq 0) \wedge (x - z = u)\}$
  $z := z + 1;$
  $y := y * z;$
  $\{(y = z!) \wedge (x - z \geq 0) \wedge (x - z < u)\}$
}
$\{y = x!\}$

De post-conditie aan het eind van de while-loop twee keer omhoog halen levert

$\{\top\}$
$y := 1;$
$z := 0;$
zolang $z <> x$ **doe** {
  $\{(y = z!) \wedge (z \neq x) \wedge (x - z \geq 0) \wedge (x - z = u)\}$
  $\{(y * (z+1) = (z+1)!) \wedge$
  $\quad (x - (z+1) \geq 0) \wedge (x - (z+1) < u)\}$
  $z := z + 1;$
  $\{(y * z = z!) \wedge (x - z \geq 0) \wedge (x - z < u)\}$
  $y := y * z;$
  $\{(y = z!) \wedge (x - z \geq 0) \wedge (x - z < u)\}$
}
$\{y = x!\}$

We moeten dus laten zien dat

$$(y = z!) \wedge (z \neq x) \wedge (x - z \geq 0) \wedge (x - z = u) \;\Rightarrow\; (y * (z+1) = (z+1)!) \wedge (x - (z+1) \geq 0) \wedge (x - (z+1) < u)$$

Eigenlijk is dit niet meer dan middelbare school-algebra, zij het met wat veel termen zodat we snel het overzicht zouden kunnen verliezen. Laten we daarom hetgeen te bewijzen is in stukjes hakken.

i) De bewering $y * (z+1) = (z+1)!$ volgt uit $y = z!$ als we links en rechts met $z+1$ vermenigvuldigen.

ii) De bewering $x - (z+1) < u$ volgt uit $x - z < u$, omdat de linkerkant van $x - (z+1) < u$ één kleiner is dan de linkerkant van $x - z < u$, en de rechterkanten hetzelfde zijn.

iii) De bewering $x - (z+1) \geq 0$ volgt uit $z \neq x$ en $x - z \geq 0$. Immers, $x - (z+1) \geq 0$ is gelijkwaardig met $x > z$ en dat wordt inderdaad door $z \neq x$ en $x - z \geq 0$ ($x \geq z$) geïmpliceerd.

Opgave 17.17 Bewijs de volledige correctheid van alle programma's uit Opgave 17.15 op blz. 212. Kijk voor elk onderdeel eerst naar de bijbehorende executietabel in het antwoord van Opgave 17.15.

17.6 Onbeslisbaarheid van terminatie

Na enige oefening lijkt het correct bewijzen van eenvoudige programma's min of meer een routinezaak. Zou het niet mooi zijn als het correct bewijzen van programma's ge-automatiseerd kan worden? Onze opdracht is dan een controle-programma te schrijven dat, gegeven een Hoare tripel, kan beslissen of dat Hoare tripel correct is. Voor statements als assignment, concatenatie en alternatief (i.e., als-dan-anders) hebben we inmiddels ervaren dat zoiets geen probleem is: we trekken de post-conditie gewoon door het programma heen naar boven. Alleen het vinden van een geschikte variant en invariant voor 'Zolang'-statements blijft moeilijk. Wordt deze moeilijkheid veroorzaakt simpelweg doordat we het overzicht verliezen, of is het vinden van een geschikte variant en invariant moeilijk omdat het, zoals voorafgaand aan Voorbeeld 17.15 op blz. 209 opgemerkt een "kunst" is?

Het vinden van geschikte varianten en invarianten blijkt een zeer diep probleem. Om hier een begin van een antwoord op te geven gaan we het Collatz-probleem bespreken. De rest van het antwoord volgt dan daarna. Bekijk het volgende programma, genaamd "Collatz":

```
{x ≥ 1}
zolang x <> 1 doe {
  als x = 0 (mod 2)
    dan x := x/2;
    anders x := 3 * x + 1;
}
{x = 1}
```

Enkele runs voor verschillende startwaarden van x zijn hieronder weergegeven:

1.
2, 1.
3, 10, 5, 16, 8, 4, 2, 1.
4, 2, 1.
5, 16, 8, 4, 2, 1.
6, 3, 10, 5, ..., 1.
7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, ..., 1.
8, 4, 2, 1.
9, 28, 14, 7, ..., 1.
10, 5, ..., 1.
11, 34, 17, 52, 26, 13, 40, 20, 10, ..., 1.
12, 6, ..., 1.
13, 40, 20, 10, ..., 1.
14, 7, ..., 1.
15, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, ..., 1.
16, 8, ..., 1.
17, 52, 26, 13, 40, 20, 10, ..., 1.
18, 9, ..., 1.
19, 58, 29, 88, 44, 22, 11, ..., 1.

Je ziet dat het programma vroeg of laat altijd lijkt te eindigen bij 1. Voor machten van twee is dit duidelijk, en ook voor andere waarden van x, zoals bijvoorbeeld "$3n+1$-voorgangers" van 2-machten, kun je wel inzien dat dit programma uiteindelijk eindigt bij 1. Sommige reeksen zijn extreem lang. Voor het getal 27, bijvoorbeeld, doorloopt het programma al 111 keer de while-loop met een maximum van $x = 9000$, voordat x uiteindelijk 1 bereikt. Voor x rond de 80 worden al maxima van x rond de 10,000 bereikt. Merk overigens op hoe de plaatsing van de puntjes ... is gekozen: als n convergeert naar een zekere $m < n$ en m zelf convergeert naar 1, dan convergeert n uiteraard ook naar 1.

De volgende vraag dringt zich nu op:

Termineert Collatz voor alle $x \geq 1$?

Deze vraag vormt een prikkelend open probleem in de wiskunde cq. informatica. Omdat het zo makkelijk te formuleren en te begrijpen is, heeft dit vraagstuk misschien wel honderdduizenden professionele en amateurwiskundigen verleid om te proberen er een definitief antwoord op te geven. Officëel staat het vermoeden dat dit programma altijd stopt bekend als *het vermoeden van Collatz*. (Ook wel: het vermoeden van Ulam, het probleem van Kakutani, het vermoeden van Thwaites, Hasse's algoritme, of het probleem van Syracuse.) Er wordt beweerd dat Lothar Collatz het probleem in 1937 introduceerde. Tot op heden is het vermoeden bevestigd noch weerlegd.[2]

Het bovenstaande programma is gemakkelijk partiëel correct te bewijzen. Echter, het vinden van een

[2] We schrijven 2018.

terminatiebewijs, meer bepaald het vinden van een geschikte variant voor (de while-loop in) het Collatz-programma is vele, vele, malen moeilijker. Immers, het vinden van een variant zou gelijk staan aan het (positief) beantwoorden van het vermoeden van Collatz.

Opgave 17.18 *i)* Leg uit dat het vinden van een partieel correctheidsbewijs van Collatz geen probleem is.

ii) Leg uit dat het vinden van een geschikte variant voor (de while-loop in) het Collatz-programma het Collatz-probleem oplost.

In de volgende sectie zal worden uitgelegd dat het vinden van geschikte varianten inderdaad een kunst is, in de zin dat het niet valt te mechaniseren. Feitelijk zal worden uitgelegd dat er geen programma bestaat, of kan bestaan, dat voor alle programma/input-combinaties kan uitmaken of het stopt.

17.7 Het stop-probleem

Het stop-probleem (Eng.: *Halting problem*) is verrassend makkelijk te formuleren, maar tegelijkertijd een ongelofelijk diep en subtiel resultaat in de informatica. Het zegt dat er geen controle-programma te schrijven is dat voor alle programma/input-combinaties kan uitmaken of het stopt of niet. Bij de meeste mensen duurt het jaren voordat ze het stop-probleem in zijn volledige diepte kunnen doorzien, dus neem de tijd om één en ander tot je te nemen.

Om het stop-probleem te introduceren, voeren we wat notatie in. Laat A een voldoend rijk alfabet zijn om een programmeertaal naar wens, J, samen met alle mogelijke inputs, I, te kunnen formuleren. J = Java en A = ASCII is prima om de gedachten te bepalen. We identificeren J voor het gemak met alle (syntactisch correcte) programma's. Dus $j \in J \Leftrightarrow j$ is een (syntactisch correct) programma in de taal J. Merk op:

$$J \subseteq I.$$

Immers, elk programma is een string. Als programma j ariteit k heeft (i.e., k input-waarden nodig heeft), dan noteren we dat met j/k.

We hebben hier overigens voor de letters J en j gekozen i.p.v. de letters Π en π, om de associatie met Java te vergemakkelijken.

Stelling 17.3 (Het stop-probleem, Turing, 1936) *Zij J een programmeertaal. Als J voldoende rijk is (en Java is dat bijvoorbeeld zeker), dan bestaat er geen programma $h \in J$ dat voor elke programma/input-combinatie $(j,i) \in J \times I$ kan uitmaken of programma j met input i stopt.*

[3] Hoe dat kan werd uitgelegd in Hoofdstuk 4 (blz. 43).

Merk op: h is twee-plaatsig en j is één-plaatsig. We kunnen dus schrijven $h/2$ resp. $j/1$.

In de loop van de geschiedenis van de informatica is het stop-probleem op verschillende manieren bewezen, van zeer formeel en omslachtig tot zeer informeel en fout. Hier volgt ons bewijs. Het is, ondanks haar informaliteit, hopelijk nog steeds correct.

Bewijs: Stel zo'n controle-programma $h \in J$ bestaat toch. (We redeneren verder en merken dan dat we op een tegenspraak uitkomen, zodat deze aanname niet houdbaar blijkt.) We mogen zonder beperking der algemeenheid aannemen dat h zó geprogrammeerd is dat deze een 1 afdrukt als $j(i)$ stopt, en 0 anders:

$$h(j,i) = 1 \;\Leftrightarrow\; \text{executie } j(i) \text{ stopt.}$$

Ga na dat deze aanname inderdaad geen wezenlijke beperking is. Dus h is een programma werkt op twee strings. De eerste string wordt opgevat als programma j, de tweede string wordt opgevat als input i, en h retourneert 1 als en slechts als j stopt op input i.

We gaan een tabel maken die aangeeft hoe h zich gedraagt. Daartoe enumereren we I.[3] Merk op dat $J \subseteq I$ zodat J meegenomen wordt in de aftelling. In de volgende tabel worden rijen als programma's, en kolommen als input geïnterpreteerd:

h	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	…
j_1	**1**	1	1	1	1	1	1	1	1	…
j_2	0	**1**	1	1	1	1	1	1	1	…
j_3	0	0	**0**	0	0	0	0	1	0	…
j_4	1	1	1	**0**	1	1	1	1	1	…
j_5	1	1	1	0	**1**	1	1	1	1	…
j_6	1	1	1	1	1	**0**	0	1	1	…
j_7	1	1	1	1	0	1	**1**	1	1	…
j_8	1	1	0	1	0	1	1	**1**	1	…
j_9	1	1	0	0	1	1	1	1	**1**	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Dus in deze tabel geldt $j_k = i_k$ voor alle $k = 1,2,3,\ldots$. We schrijven j_k, omdat we willen benadrukken dat we de string j_k op dat moment als programma te willen zien. Als willekeurige strings als programma's worden geïnterpreteerd is het natuurlijk zo dat de meeste programma's niet compileren en dus meteen stoppen. Behalve dat de tabel dan erg veel enen bevat is dat verder geen probleem.

Net zoals in Cantor's diagonaalargument (Stelling 4.4, blz 49) gaan we weer diagonalisatie toepassen. De diagonaal levert ons namelijk materiaal om een tegendraads programma q te construeren dat zich gegarandeerd anders zal gedragen dan elke $j_1, j_2, j_3, \ldots$. Meer bepaald:

> q is het programma dat zich op iedere input, i, precies *niet* zo gedraagt als programma $j = i$ op input i.

In pseudo-code:

programma $q(i)$:
als $h(i,i) = 1$
  dan stop niet
    { doe expres een oneindige loop ofzo }
  anders stop wel

Voor ieder programma j geldt dat q zich anders gedraagt dan programma j, en wel tenminste op input j. Immers, als $h(j,j) = 1$, dan zal q zelf juist niet stoppen op input j. Omgekeerd, als h bepaalt dat (j,j) niet stopt, dan zal q zelf juist *wel* stoppen op input j. Korter:

$$h(j,j) = 1 \Leftrightarrow j(j) \text{ stopt} \Leftrightarrow q(j) \text{ stopt niet.}$$

Dus $j \neq q$. Dit geldt voor alle $j \in J$, dus $q \notin J$.

We gaan nu het tegenovergestelde beargumenteren, namelijk dat $q \in J$. De letterlijke weergave hierboven is natuurlijk geen Java, het is slechts pseudo-code ("**zolang** true **doe** x=0", "stop niet", "stop wel", etc.). We hadden gekozen voor Java. Is q te schrijven in Java? Met enige verbeelding is in te zien dat q inderdaad geschreven kan worden in Java. Code voor h hadden we al (dat was immers aangenomen). Wat rest is de pseudo-code om te schrijven naar Java en vanuit daar h aan te roepen. We mogen dus concluderen dat $q \in J$.

We komen uit op een tegenspraak, immers eerder was al geconcludeerd dat $q \notin J$. Omdat we geloven dat al onze andere uitgangspunten en redeneringen correct zijn, kunnen we niet anders dan concluderen dat onze oorspronkelijke aanname niet waar kan zijn. Er kan dus geen controle-programma $h/2$ bestaan dat van een willekeurig programma $j/1$ en een willekeurige input i kan controleren of $j(i)$ stopt. □

Opgave 17.19 Onderzoek of q losgelaten op q, i.e, $q(q)$ stopt. Eerst volgens h, daarna volgens de specificatie van q.

Stelling 17.3 impliceert allerlei andere onmogelijkheids-stellingen. Voor elk van de volgende beweringen (met i en j willekeurig) bestaan geen controleprogramma's die kunnen uitmaken of deze beweringen waar zijn.

1. Programma $j/1$ stopt op input i. (Dit is Stelling 17.3.)
2. Programma $j/1$ stopt altijd. (Dus voor alle input i stopt het programma. Dit wordt het uniforme stop-probleem genoemd.)
3. Er bestaat een input i voor programma $j/1$ zo dat j stopt. (Het existentiële stop-probleem.)
4. Programma $j/0$ stopt. (We bekijken dan alleen programma's met ariteit nul.)
5. Programma $j/0$ schrijft niet meer dan niet dan k karakters.
6. Programma $j/0$ gebruikt niet meer dan niet dan k KB aan geheugen.
7. Programma $j_1/1$ en $j_2/1$ stoppen op dezelfde input i.
8. Programma $j/0$ zal symbool $a \in A$ afdrukken.
9. Programma $j/0$ zal instructie s uitvoeren.
10. Programma $j'/0$ is het kortste programma dat hetzelfde doet als programma $j/0$.
11. Programma's $j/0$ en $j'/0$ gedragen zich hetzelfde. (Het equivalentie-probleem.)

Laten we wat praktischer worden en opgaven met Java-programma's doen. Laten we een Java-programma dat precies n input-strings nodig heeft om te kunnen werken, een *Java/n programma* noemen. De ariteit van een Java/n-programma is dus n. Een Java-programma dat bijvoorbeeld gemeenschappelijke klinkers uit twee woorden haalt, zal typisch een Java/2-programma zijn. Immers, de invoer zal waarschijnlijk bestaan uit twee woorden.

Een consequentie van de onbeslisbaarheid van het stop-probleem is dat er geen programma (of mechanische procedure of algoritme) bestaat dat voor alle combinaties van Java/1-programma's en input-strings kan beslissen of dat Java-programma met die specifieke input stopt. Immers, Java is expressief genoeg om de gebruikelijke programma-constructies uit te drukken.

Voorbeeld 17.18 (Het inputloze stop-probleem) Toon met een reductie-argument aan dat er zélfs geen programma (of mechanische procedure of algoritme) bestaat dat voor alle Java/0-programma's (i.e., input-loze Java-programma's) kan beslissen of het stopt.

De uitwerking is als volgt: we stellen dat er toch zo'n controle-programma bestaat, en laten dan zien dat met deze aanname dan ook een controle-programma (of controle-procedure of test-algoritme) bestaat dat voor elk Java/1-programma, j, en voor elke string, i, kan beslissen of $j(i)$ stopt.) Dat laatste kan niet, dus de oorspronkelijke aanname is onhoudbaar.

Dus stel dat er toch een programma (of mechanische procedure of algoritme) bestaat dat voor alle Java/0-programma's (i.e., input-loze Java-programma's) kan beslissen of het stopt. We laten nu zien dat we met behulp van dit programma het stop-probleem zouden kunnen oplossen. Laat j/i een willekeurige programma/input-combinatie zijn waarvan moet worden bepaald of het stopt. Laat Java-programma j' ontstaan uit j/i door de lees-statement, i.e, zo iets als

```
BufferedReader br =
   new BufferedReader(
      new InputStreamReader(System.in));
String x = br.readLine();
```

te vervangen door een expliciete toekenning van x met i, i.e., zoiets als

```
String x = i;
```

Eenvoudig is na te gaan dat $j(i)$ stopt als en slechts als j' stopt. We zouden het stop-probleem dus kunnen oplossen door (j,i)-combinaties automatisch te laten te omschrijven naar een inputloos Java-programma j' waarvan we dan,

volgens de aanname, wél zouden kunnen beslissen of het stopt. Dat is in tegenspraak met de onbeslisbaarheid van het stop-probleem, dus het stop-probleem voor input-loze Java-programma's is daarmee ook onbeslisbaar.

We zien een algemene bewijs-strategie: wil je laten zien dat een probleem P onbeslisbaar is, laat dan zien dat het bestaan van een beslis-algoritme voor P een beslis-algoritme voor het stop-probleem (of een ander bekend onbeslisbaar probleem) impliceert.

Opgave 17.20 Welke van de volgende problemen zijn beslisbaar?

i) Bepaal of een willekeurig Java/0-programma in 10,000 of minder instructies stopt.

ii) Bepaal of een willekeurig Java/0-programma 10,000 of minder karakters afdrukt.

a. Beide niet.

b. Alleen *i)*.

c. Alleen *ii)*.

d. Zowel *i)* als *ii)*.

Graag je antwoord motiveren. (We herinneren je er aan dat de onbeslisbaarheid van een probleem, zeg P, kan worden aangetoond door, voor een bekend onbeslisbaar probleem, zeg H, een reduceeralgoritme (reduceervoorschrift) te geven dat H reduceert tot het oorspronkelijk probleem P.)

Opgave 17.21 Bewijs dat de volgende problemen onbeslisbaar zijn:

1. (Het print-probleem.) Bepaal voor een willekeurig programma j en een willekeurig symbool a of j ooit a zal afdrukken.
2. (Het uniforme stop-probleem.)
3. (Het programma-equivalentie probleem.)

Veel onbeslisbaarheidsproblemen kunnen in één keer worden bewezen: in 1953 bewees de Amerikaanse wiskundige Henry Rice dat alleen zogenaamde *triviale* eigenschappen van programma's beslisbaar zijn, waarbij een eigenschap gedefinieerd is als *triviaal* als en slechts als de betreffende eigenschap voor alle programma's *wel* geldt of juist voor alle programma's *niet* geldt.

Niet alles is verloren: er bestaat bijvoorbeeld wél een controle-programma $w/1$ dat kan bepalen of een willekeurig ander programma, j, iets afdrukt (of wegschrijft naar een bestand, of überhaupt iets veranderd aan de omgeving). Het programma j mag elke ariteit bezitten.

$w(j) = 1 \Leftrightarrow$
j drukt op enig moment een symbool $a \in A$ af.

[4] Zie Wikipedia: "paging".

Hoe? In grote lijnen doet w het volgende. Zodra w het programma j met input $i_1, \ldots, i_n$ krijgt aangeboden voert w het programma j uit (w is dus een interpreter). Nu zijn er verschillende mogelijkheden:

1. Het programma j compileert niet of crasht zonder dat het ooit wat heeft afgedrukt. Het programma w kan dan een 0 afdrukken en stoppen.
2. Op een gegeven moment drukt j wat af. Het programma w kan dan een 1 afdrukken en stoppen.
3. Het programma j blijft maar doorlopen zonder ooit iets af te drukken.

Het laatste geval hoeft ook niet problematisch voor w te zijn, want we kunnen w zo schrijven dat het de inhoud van alle registers van j monitort. Laten we alle mogelijke registerconfiguraties *toestanden* noemen. Omdat er eindig veel registers zijn, zijn er ook maar eindig veel toestanden mogelijk. Als j maar blijft doorlopen zonder ooit iets af te drukken (en zonder iets te lezen uit de omgeving, immers alle input was bij aanvang gegeven), dan zal j bij eindig veel toestanden uiteindelijk in een eerder bezochte toestand terecht moeten komen. Als j op dat moment nóg niets heeft afgedrukt zit j overduidelijk in een loop gedurende waarin j nooit iets afdrukt. Het programma w kan j dan afbreken en naar waarheid een 0 afdrukken. □

Opgave 17.22 1. Beargumenteer dat als een programma dat niet mag schrijven, of überhaupt iets aan de omgeving mag veranderen, een symbool $a \in A$ afdrukt, dat het dan oneindig vaak dat symbool a afdrukt.

2. Beargumenteer dat, als een programma wél mag schrijven, en we verder mogen aannemen dat de schrijfruimte (denk aan schijfruimte) onbeperkt is (denk aan onbeperkt mogen "pagen")[4] de bovenstaande beslisbaarheidsredenering niet meer opgaat.

3. Waarom is de beslisbaarheid van dit probleem niet in strijd met het print-probleem (Opgave 17.21)?

17.8 De stop-functie

In Hoofdstuk 5 werd een existentieel bewijs gegeven voor het bestaan van een onberekenbare functie. Een onsympathieke eigenschap van een existentieel bewijs is dat het non-constructief is: in dit geval wordt er geen onberekenbare functie *gegeven*, er wordt slechts aangetoond dat er tenminste één *bestaat*.

De constructivisten onder jullie kunnen rustig worden. Op dit moment hebben we voldoende gereedschap om zelf een onberekenbare functie $f : \mathbb{N} \to \mathbb{N}$ te construeren. Deze functie is gebaseerd op de onmogelijkheid van het bestaan van een programma

$$H : J \to \{0, 1\}$$

dat voor inputloze programma's $j/0 \in J$ kan bepalen of j stopt. (Zie Opgave 17.18, blz. 216. Overigens, het feit dat het domein van H programma's van elke ariteit bevat is geen probleem. We kunnen afspreken dat H ongedefinieerd is op programma's van ariteit 1 en hoger.)

De functie die we zoeken is H. Het probleem is dat H het verkeerde domein heeft. We zoeken namelijk een onberekenbare functie van getallen naar getallen in plaats van een onberekenbare functie van programma's naar getallen.

Om een onberekenbare functie $h : \mathbb{N} \to \mathbb{N}$ te krijgen moeten we op de één of andere manier het domein van de onberekenbare functie H (dat is J) omzetten naar natuurlijke getallen die *op een berekenbare manier* corresponderen met J. We zoeken dus een functie $x : \mathbb{N} \to J$ die berekenbaar is, zodat we een functie-compositie

$$\mathbb{N} \underbrace{\xrightarrow{x}}_{\text{berekenbaar}} J \underbrace{\xrightarrow{H}}_{\text{onberekenbaar}} \{0,1\}.$$

kunnen maken. Zo'n functie x is er: op blz. 60 werd aangegeven hoe op een berekenbare manier programma's éénduidig kunnen worden genummerd door middel van een zogenaamde Gödelnummering $G : J \to \mathbb{N}$. Ook werd daar aangegeven dat de inverse $G^{-1} : \mathbb{N} \to J$ ook berekenbaar is. Uit Opgave 3.32 (blz. 41) volgt dat G^{-1} een links-inverse is van G, dat wil zeggen dat $G^{-1} \circ G$ de identieke functie I op J is, i.e., de functie waarvoor geldt $I(j) = j$, voor alle $j \in J$.

De gezochte functie is dus misschien $x = G^{-1}$. (We zeggen "misschien" omdat we nog niet weten of deze functie gaat doen wat we er van verwachten.) We krijgen het diagram

$$\mathbb{N} \underbrace{\xrightarrow{?}}_{\text{berekenbaar}} G^{-1} \underbrace{\xrightarrow{H}}_{\text{onberekenbaar}} \{0,1\}.$$

Intuïtief voel je nu al aan dat de functie

$$h = H \circ G^{-1} : \mathbb{N} \to \mathbb{N}$$

onberekenbaar is. Stel, om een tegenspraak te krijgen, dat h tóch berekenbaar is. Uit Opgave 5.8, blz. 63 volg dat de compositie van twee berekenbare functies ook berekenbaar is. Dus de functie $h \circ G : J \to \mathbb{N}$ is ook berekenbaar. Maar

$$h \circ G = (H \circ G^{-1}) \circ G = H \circ (G^{-1} \circ G) = H \circ I = H,$$

en H is nou juist onberekenbaar. De aanname dat h berekenbaar is, blijkt onhoudbaar. □

Terug naar Floyd-Hoare logica. Voor de Floyd-Hoare logica impliceert bovenstaand resultaat dat er geen systematische methode (of algoritme, dat is hetzelfde) kan bestaan, dat voor elk willekeurig Hoare/Java-tripel kan verifiëren of deze correct is.

Er valt meer over Floyd-Hoare logica te zeggen dan wij hier gedaan hebben. Voor meer informatie verwijzen we naar [Van Amstel 1984] (voor correctheidsbewijzen met betrekking tot Pascal) en [Gordon 1988] (hoofdstukken 1 en 2 geven een goede inleiding in Floyd–Hoare logica; het boek gaat ook in op het mechaniseren van correctheidsbewijzen). Een standaardwerk over correctheidsredeneringen voor imperatieve programma's is [Gries 1983].

Hoofdstuk 18

Dynamische logica

De Floyd–Hoare logica uit de vorige paragraaf was tamelijk geschikt voor het bestuderen van de correctheid van **zolang**-programma's. Over een aantal andere zaken kan deze logica echter bijzonder weinig zeggen:

– *Termineert* een gegeven programma?

– Zijn twee verschillende programma's *equivalent*?

– Is een willekeurig programma *deterministisch* of niet?

De laatste vraag heeft betrekking op mogelijke keuzevrijheid binnen een programma (in zekere zin: oude metafysische problemen in een nieuw jasje). Het programma kan zich bij gelijke data verschillend gedragen wanneer het meerdere keren "gedraaid" wordt; in dat geval noemen we het programma *indeterministisch*.[1] Dit kan zich niet voordoen bij **Impertaal**: **zolang**-programma's zijn deterministisch.

PROLOG programma's met een gefixeerde bewijsstrategie (bij voorbeeld 'eerst links en naar beneden') zijn ook deterministisch: het verwerkingsproces ligt dan geheel vast. Wanneer we een PROLOG programma—bij voorbeeld het programma dat de familierelaties binnen ons koningshuis beschrijft—door een procedurele bril beschouwen, en bovendien de PROLOG bewijsstrategie even uitzetten, dan kunnen we zeggen dat zo'n programma indeterministisch is wanneer het opdrachten toelaat die op meer dan één manier kunnen worden uitgevoerd. De vraag 'Is X een zus van Beatrix?' ziet er procedureel uit als een opdracht: 'Noem een zus van Beatrix!' Wanneer er meerdere mogelijke antwoorden zijn ('Margriet', 'Irene', 'Christina') dan noemen we het programma *indeterministisch*. We vergeten dan dus even dat de PROLOG bewijsstrategie in feite vastlegt welk antwoord het eerst wordt gegeven. Op dezelfde manier worden PROLOG predikaten die op meer dan een manier waar kunnen worden gemaakt *indeterministisch* genoemd; de PROLOG predikaten waarbij dat niet zo is heten *deterministisch*. Één en ander maakt duidelijk dat het verband met het metafysische begrip 'indeterminisme' nu ook weer niet zo direct is.

Maar er zijn moderne toepassingen zoals verspreide berekeningen—op een stel aan elkaar verbonden computers die nooit alles van elkaar kunnen weten—die naar hun aard indeterministisch zijn. We zullen ons hier echter beperken tot **Impertaal** programma's, die zoals gezegd altijd deterministisch zijn.

De vraag naar de equivalentie van programma's is met de nodige extra theorievorming nog wel gedeeltelijk binnen de correctheidslogica te beantwoorden, maar de eerste vraag (naar mogelijke terminatie) onttrekt zich daar geheel aan. De Floyd–Hoare logica is te weinig flexibel om behalve correctheid ook andere wezenlijke kenmerken van het gedrag van programma's te onderzoeken. Daarom presenteren we in deze paragraaf een rijkere logische taal waarin we de programma's zelf kunnen representeren.

De zogenaamde dynamische logica (DL) beoogt de veranderingen te beschrijven die een computerprogramma bewerkstelligt: na het draaien van een programma ziet de wereld er anders uit dan daarvoor. Hierin speelt het begrip *toestand* van de machine weer een belangrijke rol: een programma of opdracht kan de machinetoestand veranderen. Begin- en eindtoestand zullen bij het uitvoeren van een programma dus in de regel verschillen. Met andere woorden, het programma maakt de eindtoestand *bereikbaar* vanuit de begintoestand. Een model voor een programma π bevat dus een verzameling toestanden en een toegankelijkheidsrelatie T_π daartussen. In begin- en eindtoestand zullen uiteraard verschillende formules waar gemaakt worden; vergelijk de begin- en eindvoorwaarden uit de Floyd–Hoare logica.

Dit doet u wellicht denken aan de semantiek van de modale logica, en dat was precies de bedoeling. De toestanden zijn in feite wat de werelden waren in de intensionele modellen. We kunnen nu dus spreken over beweringen die *noodzakelijk waar* zijn volgens het programma. 'Noodzakelijk waar' wil hier zeggen: waar in elke toestand die ontstaat als het programma gestopt is. Er is wel een kleine notationele complicatie: bij elk programma π hoort nu een verschillende noodzakelijksoperatie $\Box$. Het vierkantje indiceren als $\Box_\pi$ of $\boxed{\pi}$ wordt echter lastig als het programma een zekere omvang krijgt. Daarom wordt het vierkantje in de regel opengebroken: de notatie wordt dan $[\pi]$. En net zo voor de tegenhanger van 'mogelijkerwijs' in dynamische logica: $\langle\pi\rangle$. De definitie van 'mogelijk volgens π' is als vanouds: $\langle\pi\rangle\varphi = \neg[\pi]\neg\varphi$. Ofwel: in minstens één van de toestandsovergangen die π kan bewerkstelligen is φ waar.

De waarheidsdefinitie van dynamische formules is overeenkomstig de semantiek van de modale logica. ψ is waar in b wordt nu genoteerd als $b \models \psi$, vergelijk de predikatenlogische notaties $V_b(\psi) = 1$ of $\models \psi[b]$. De cruciale waarheidsvoorwaarden van dynamische formules luiden (b en e zijn willekeurige toestanden):

– $b \models [\pi]\varphi \Leftrightarrow$ voor elke e zodat $bT_\pi e$ geldt dat $e \models \varphi$;

– $b \models \langle\pi\rangle\varphi \Leftrightarrow$ er is een e waarvoor $bT_\pi e$ en $e \models \varphi$.

We kunnen nu correctheidsbeweringen binnen de logica zelf doen. Dat is anders dan in de Floyd–Hoare logica die

[1] In de praktijk spreekt men meestal van *non-deterministisch* (GV).

weliswaar predikaatlogische formules gebruikt, maar zelf geen predikaatlogische theorie is: de correctheidsbewering $\vdash \{\varphi\}\pi\{\psi\}$ is een *metalogische* formulering; het kan nog het best beschouwd worden als een speciale gevolgtrekkingsrelatie $\overset{\pi}{\Rightarrow}$ tussen φ en ψ. Dezelfde bewering kunnen we zonder verdere poespas weergeven in dynamische logica: $\varphi \to [\pi]\psi$.

Voorbeeld 18.1 Beschouw het volgende programma π:

```
begin
  k := 0;
  j := x;
  zolang j > 0 doe
    begin k := k + x; j:= j - 1 einde
einde
```

Om te zien wat het programma doet kiezen we een bepaalde waarde van x, zeg $x = 3$ en "draaien" het programma waardoor de computer achtereenvolgens de onderstaande serie toestanden (weergegeven door tripels $\langle x, j, k\rangle$) doorloopt:

$$\langle 3,3,0\rangle, \langle 3,2,3\rangle, \langle 3,1,6\rangle, \langle 3,0,9\rangle.$$

Het programma telt voor elke aan x gegeven waarde $x-1$ keer x bij x op, kortom het kwadrateert x, mits x niet negatief is. In de Floyd–Hoare calculus zouden we dit als $\{x \geq 0\}\pi\{k = x^2\}$ noteren; in dynamische logica luidt dezelfde specificatie: $x \geq 0 \to [\pi](k = x^2)$.

De zojuist geformuleerde correctheidsbewering steunt echter nog steeds op de aanname dat het programma termineert. Immers $[\pi]\varphi$ is waar in een bepaalde toestand als er helemaal geen π-toegankelijke toestanden zijn. Dit is niet het geval voor een formule als $\langle\pi\rangle\varphi$: deze wordt alleen waargemaakt als er een overgang is naar een toestand waarin φ waar is. Dus drukt in het bovenstaande voorbeeld de formule $x \geq 0 \to \langle\pi\rangle(k = x^2)$ zowel terminatie als correctheid (zogenaamde *totale correctheid*) uit. Meer in het algemeen kunnen we stellen dat de formule $\langle\pi\rangle\top$ *terminatie* van een programma π weergeeft. Kortom:

> $\langle\pi\rangle\top$ is waar $\Leftrightarrow$ T_π is voortzettend $\Leftrightarrow$ π termineert.

We zien aan dit voorbeeld eveneens dat $\langle\pi\rangle\varphi$ een sterkere bewering kan zijn dan $[\pi]\varphi$. Het is zelfs zo dat $\langle\pi\rangle\varphi \to [\pi]\varphi$ voor **Impertaal**-programma's π een *geldige* formule is. Deze (voor modale logica uitzonderlijke) situatie wordt veroorzaakt door het feit dat zulke programma's *deterministisch* zijn: als π termineert dan is de eindtoestand bepaald door de begintoestand. Anders gezegd: vanuit elke toestand is er steeds hoogstens één toestand toegankelijk. Samengevat:

> $\langle\pi\rangle\varphi \to [\pi]\varphi$ is geldig voor elke φ $\Leftrightarrow$ T_π is een partiële functie $\Leftrightarrow$ π is deterministisch.

Ook over equivalentie van programma's kunnen we nu uitspraken doen. Hier zijn meerdere zienswijzen mogelijk. Een heel stringente eis aan equivalentie van programma's is te eisen dat ze altijd een gelijke interpretatie (dat wil zeggen: toegankelijkheidsrelatie) krijgen.

> $[\pi_1]\varphi$ en $[\pi_2]\varphi$ zijn logisch equivalent voor elke φ $\Leftrightarrow$ $T_{\pi_1} = T_{\pi_2}$ in elk model $\Leftrightarrow$ π_1 en π_2 zijn (sterk) equivalent.

Deze eis kan echter overdreven sterk blijken te zijn, getuige het volgende voorbeeld.

Voorbeeld 18.2 Laat π het programma zijn uit voorbeeld 18.1 dat positieve getallen kwadrateert. We amenderen het programma enigszins om ook negatieve getallen te kwadrateren. Laat daarom π_1 het volgende programma zijn:

```
begin
  (als x < 0 doe x := -x);
  k := 0;
  j := x;
  zolang j > 0 doe
    begin k := k + x; j:= j - 1 einde
einde
```

Omdat π en π_1 voor negatieve x een ander resultaat leveren zullen we deze programma's eigenlijk niet equivalent willen noemen. π_1 is daarentegen intuïtief gesproken wel equivalent met het eenvoudige programma π_2:

```
k := x * x
```

Echter π_1 en π_2 zijn volgens de definitie niet sterk equivalent: ze maken niet dezelfde formules waar; bijvoorbeeld $[\pi_1](j = 0)$ is geldig, maar $[\pi_2](j = 0)$ zeker niet.

Zoals voorbeeld 18.2 duidelijk maakt is het beter het begrip *equivalentie* wat af te zwakken. Voor een willekeurige formule φ stellen we:

> $[\pi_1]\varphi$ en $[\pi_2]\varphi$ zijn logisch equivalent $\Leftrightarrow$ π_1 en π_2 zijn equivalent ten opzichte van φ.

We moeten nog een laatste hobbel nemen voordat we daadwerkelijk programma's kunnen interpreteren. De situatie is nu namelijk dat er voor bijna elk programma π een verschillende toegankelijkheidsrelatie T_π moet komen, en hoe kunnen we zo ooit inzicht verwerven in de werking van programma's? Dit probleem is echter makkelijk oplosbaar: we kunnen T_π opbouwen uit deelrelaties die corresponderen met de opdrachten waaruit π bestaat. Voor **Impertaal** gaat dat zo:

- als $\pi =$ **begin** $\pi_1;\cdots;\pi_n$ **einde**, dan $T_\pi = T_{\pi_1} \bullet \cdots \bullet T_{\pi_n}$.
- als $\pi =$ **(als** φ **dan** α**)**, dan $T_\pi = \{\langle b,e\rangle \mid b \models \varphi \Rightarrow \langle b,e\rangle \in T_\alpha \wedge b \not\models \varphi \Rightarrow b = e\}$.
- als $\pi =$ **(als** φ **dan** α **anders** β**)**, dan $T_\pi = \{\langle b,e\rangle \mid b \models \varphi \Rightarrow \langle b,e\rangle \in T_\alpha \wedge b \not\models \varphi \Rightarrow \langle b,e\rangle \in T_\beta\}$.

– als $\pi =$ **zolang** φ **doe** α, dan
$T_\pi = \{\langle b,e\rangle \mid e \not\models \varphi$ en óf $b = e$ óf er bestaan toestanden $c_1,\ldots,c_k$ waarvoor φ waar is, $c_1 = b$ en voor iedere $i : \langle c_i, c_{i+1}\rangle, \langle c_k, e\rangle \in T_\alpha\}$.

Toelichting: de eerste clausule gebruikt de operatie $\bullet$ van *samenstelling van relaties*, die gedefinieerd wordt door:

$$T_\alpha \bullet T_\beta = \{\langle b,e\rangle \mid \text{ er is een } c \text{ zodat } \langle b,c\rangle \in T_\alpha \wedge \langle c,e\rangle \in T_\beta\}.$$

Merk op deze samenstelling een generalisatie is van *compositie* van functies wanneer we functies als relaties opvatten, zij het dat de notatie andersom werkt (dus $f \bullet g = g \circ f$). Merk verder op dat $\bullet$ een *associatieve* operatie is: de volgorde waarin we het "product" berekenen doet er niet toe, we hoeven dus geen haakjes te gebruiken.

De middelste clausules behoeven weinig commentaar, maar de laatste des te meer. De interpretatie van de **zolang**-opdracht dient als volgt gelezen te worden: of φ is onwaar in een begintoestand, α wordt niet toegepast, en de toestand van de automaat blijft onveranderd, of φ is wel waar en α wordt een aantal keren toegepast totdat φ onwaar wordt.

Opgave 18.1 Wat gaat er mis als we voor de bovenstaande **als dan** clausule de volgende voor de hand liggende definitie hadden gekozen:

$$T_\pi = \{\langle b,e\rangle \mid b \models \varphi \Rightarrow \langle b,e\rangle \in T_\alpha\}.$$

Overigens zien we bij bovenstaande clausules de wisselwerking tussen logica en programmeertaal: logische formules verschijnen als Boolese uitdrukkingen in de condities van de programmeeropdrachten, en programma's treden op binnen dynamische operatoren. De notatie van uitdrukkingen of formules zal daarom enige variatie vertonen: binnen het programma-deel van de taal wordt gebruik gemaakt van `niet`, `en` en `of`, binnen het logica-deel van $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. In semantische interpretaties en logische axioma's verdonkeremanen we dit verschil vaak.

Wat bij al deze interpretaties (behalve misschien de eerste) verder in mindere of meerdere mate verwaarloosd wordt is het procesmatige karakter van het draaiende programma. De semantiek van een opdracht geeft een beschrijving alsof de hele opdracht verwerkt is, en niet hoe het dit stukje bij beetje doet. Voor het resultaat maakt dit natuurlijk niets uit.

We weten nu hoe we T_π kunnen opbouwen uit relaties die aan afzonderlijke opdrachten verbonden zijn, maar nog niet wat de basisstap van deze recursieve definitie is. Hiervoor zijn meerdere mogelijkheden. In de zogenaamde propositionele Dynamische Logica (pDL) werkt men zoals uit de naam blijkt met de taal van de propositielogica verrijkt met een programmeertaal die binnen de modale operatoren kan verschijnen. Omdat er in het puur logische gedeelte van de taal dan alleen met atomaire formules en connectieven en niet met variabelen en kwantificatie gewerkt wordt, ligt het voor de hand iets dergelijks binnen het programmeerdeel te doen. Men kiest dan naar analogie met de verzameling propositieletters een verzameling *atomaire programma's* $a, a', a'', \ldots$ ("programmaletters"). In de semantiek krijgt elk atomair programma a een willekeurige (deterministische) relatie T_a toegekend.

Eerder hebben we opgemerkt dat **Impertaal** deterministisch werkt en gesuggereerd dat het dan ook aanbeveling verdient de semantiek hiernaar in te richten.[2] We dienen nu nog wel te controleren dat de voorgestelde modellen inderdaad deterministisch zijn, dat wil zeggen dat er voor elk programma π en toestand b hoogstens één toestand e kan bestaan zodat $bT_\pi e$, uitgaande van het gegeven dat dit voor atomaire programma's per definitie het geval is. De sleutelvraag wordt nu: bewaren de interpretatieclausules determinisme? Uiteraard schreeuwt deze vraag om een inductief bewijs. We controleren een paar stappen expliciet.

Voorbeeld 18.3

opeenvolging Laat $\pi =$ **begin** $\pi_1;\cdots;\pi_n$ **einde**, en stel dat $T_{\pi_1},\ldots,T_{\pi_n}$ alle deterministisch zijn. Dan ook $T_\pi = T_{\pi_1} \bullet \cdots \bullet T_{\pi_n}$. Want kies een willekeurige toestand b, dan is er maar hoogstens één b_1 zodat $bT_{\pi_1}b_1$. Vanuit die eventuele b_1 is weer 0 of 1 b_2 met $b_1T_{\pi_2}b_2$, enzovoorts. Dus kan er voor gegeven π en b maar hoogstens 1 T_π-bereikbare toestand b_n zijn.

als–dan Laat $\pi =$ **(als** φ **dan** α**)**, en stel T_α is deterministisch. Kies een willekeurige toestand b. Dan ofwel $b \models \varphi$ en dan is er maximaal 1 T_π-bereikbare e wanneer $\langle b,e\rangle \in T_\alpha$, ofwel $b \not\models \varphi$ en dan is $e = b$ per definitie de enige T_π-bereikbare toestand.

Opgave 18.2 Bewijs zelf behoud van determinisme voor de **als–dan–anders**- en de **zolang**-stap.

Er zijn in de pDL prachtige theoretische resultaten geboekt (zie bijvoorbeeld [Harel 1984] voor het bewijs van volledigheid en beslisbaarheid) maar voor de meeste toepassingen is pDL toch ongeschikt. Er zijn nauwelijks echte programma's te bedenken die adequaat worden weergegeven in pDL; een pDL-formule als

`[zolang p doe a]`q

zal weinig realiteitswaarde bezitten omdat er voor echte programma's in de regel wel een koppeling moet zijn tussen de `p` en de `a`; kijk er de **Impertaal** voorbeelden maar eens op na.

Het alternatief is uiteraard om binnen atomaire programma's individuele variabelen en constanten (en eventueel ook functieconstanten) toe te laten. Het gebruik van *termen* (in logische zin) is volstrekt natuurlijk omdat ze ook—soms zelfs onder dezelfde naam—als type in de

[2] Logisch gezien is een indeterministische semantiek overigens best mogelijk, maar voor een eerste kennismaking lijkt dit didactisch minder gewenst.

programmeertaal voorkomen. Een atomaire opdracht is daarmee een *toekenningsopdracht* geworden. Deze combinatie van dynamische operatoren en predikatenlogica wordt wel gekwantificeerde dynamische logica genoemd; wij zullen deze combinatie gewoon aanduiden als 'dynamische logica'.

Logisch betekent een toekenningsopdracht dat er een nieuwe waarde aan een variabele wordt toebedeeld. Het enige wat er bij een toekenning mogelijk in de machinetoestand verandert is dus de *bedeling* die aangepast moet worden opdat de variabele de nieuwe waarde krijgt; precies wat we in § 11 deden om de kwantoren te interpreteren. Maar als bij atomaire opdrachten alleen de bedeling verandert, dan geldt dit vanwege de gegeven recursieve opbouw ook voor willekeurige opdrachten en programma's. Om die reden worden de toestanden in het model vanaf hier gezien als bedelingen. De interpretatie van de toekenningsopdracht is dus:

– als π de opdracht $v := t$ is dan $R_\pi = \{\langle b, e\rangle \mid e = b(v|W_b(t))\}$.

Verder brengen we in herinnering dat W_b gedefinieerd wordt door:

– $W_b(u) = b(u)$ voor variabelen u;
– $W_b(c) = I(c)$ voor constanten c;
– $W_b(f t_1 \ldots t_n) = I(f)(W_b(t_1), \ldots, W_b(t_n))$ voor n-plaatsige functieconstanten f.

Voor het gegeven fragment van **Impertaal** betekent dit dat we de rekenkundige bewerkingen (**+**,**–**, $*$) en relaties ($=, <, >, <=, >=$) in het model moeten verdisconteren. Dat kan natuurlijk op allerlei niet bedoelde manieren, maar we beperken ons hier tot zogenaamde *rekenkundige modellen*. In zo'n standaardmodel kiezen we een geschikt rekenkundig domein (meestal $\mathbb{N}$, $\mathbb{Z}$ of $\mathbb{Q}$) en een vaste interpretatie I van getallen (bij voorbeeld $I(\mathbf{10}) = 10$), operaties ($I(*) = \cdot$) en relaties ($I(<=) = \leq$).

Voorbeeld 18.4 Neem als domein van het model de verzameling gehele getallen $\mathbb{Z}$. Laat I de standaardinterpretatie zijn. Dan zijn bijvoorbeeld de volgende beweringen waar in dit model:

$(x = 4) \to \langle$`x := x + 1`$\rangle(x = 5)$
$\langle$`zolang x > 0 doe x := x - 1`$\rangle(x \leq 0)$
`[zolang x > 0 doe x := x + 1]`$(x \leq 0)$
$\forall x \forall y$`[zolang niet x = y doe x := x + 1]`$(x = y)$.

We controleren de eerste bewering. Stel de bewering is niet waar, dan is er een bedeling b zodat $V_b(x = 4) = 1$ en $V_b(\langle$`x := x + 1`$\rangle(x = 5)) = 0$. Uit het eerste volgt $b(x) = 4$. Als $e = b(x|b(x) + 1)$, en dus $e = b(x|W_b(x + 1))$, dan $\langle b, e\rangle \in T_{\mathtt{x:=x+1}}$ en $e(x) = 5$, maar dan ook $V_e(x = 5) = 1$, wat een tegenspraak oplevert.

De waarheid van de tweede bewering is ook eenvoudig na te rekenen (bedelingen die aan x een positieve waarde toekennen worden door de **zolang**-opdracht zo aangepast dat x de waarde 0 krijgt, en bedelingen waarvoor x negatief of 0 is blijven onveranderd).

De derde bewering is minder vanzelfsprekend. Ze lijkt zelfs onwaar als we een bedeling kiezen die aan x een positieve waarde toekent. Maar in dat geval levert de interpretatie van de **zolang**-opdracht geen toegankelijke toestand op, en in het andere geval blijft de bedeling weer onveranderd, en dus is $x \leq 0$ waar in iedere toegankelijke toestand.

Opgave 18.3 Bewijs de waarheid van de laatste van de vier beweringen in bovenstaand voorbeeld.

Hoewel de dynamische logica vooral semantisch is gemotiveerd kunnen we haar ook vanuit axiomatisch gezichtspunt bekijken: welke afleidingsprincipes liggen eraan ten grondslag? Allereerst gebruiken we alle axioma's, afleidingsregels (en daarmee alle stellingen en afgeleide regels) van de predikatenlogica. Daarnaast herhalen we de principes die ook geldig waren in de (zogenaamde normale) modale predikatenlogica die we in dit dictaat gepresenteerd hebben:

Axioma 18.1 (K, "verdeling")

$$\vdash [\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi).$$

Afleidingsregel 18.1 (N, "necessitatie")

$$\vdash \varphi \Rightarrow \vdash [\pi]\varphi.$$

Deze principes zijn immers ook semantisch geldig, omdat ze op Kripke modellen altijd waar zijn. Daarnaast zijn er principes die kenmerkend zijn voor dynamische logica, en voor een deel zelfs afhangen van de gekozen programmeertaal. Voor een gestructureerde, deterministische taal als **Impertaal** geldt zoals gezegd het axioma:

Axioma 18.2 (determinisme)

$$\vdash \langle\pi\rangle\varphi \to [\pi]\varphi.$$

Vergeleken met de Floyd–Hoare calculus is de toekenningsopdracht op zichzelf onproblematisch:

Axioma 18.3 (toekenning)

$$\vdash [v := t]\varphi \to [t/v]\varphi.$$

Overigens zorgt de dynamische toekenningsoperator wel voor een andere complicatie: ze bindt net als kwantoren en de lambda-operator variabelen die binnen haar bereik vallen. Wat dat betreft gedraagt $[v := t]$ zich precies als bij voorbeeld $\exists v$. Dat er echt sprake is van binding kunnen we onder meer zien aan het feit dat substitutie binnen het bereik van de toekenningsoperator aan dezelfde gevaren onderhevig is als binnen het bereik van een kwantor. Analoog aan § 14.2 merken we op dat

$$\forall y[\mathtt{x := y + 1}](x \neq y) \to [x/y][\mathtt{x := y + 1}](x \neq y)$$

geen geldige formule is.

Opgave 18.4 Voer de substitutie uit en toon vervolgens aan dat formule (0) niet semantisch geldig is.

Een verschil tussen $[x := t]$ en $\exists x$ is dat een toekenningsopdracht zelf weer variabelen kan introduceren en die voorkomens mogen best vrij zijn. Ter illustratie beschouwen we de ware DL-formule

$$\forall x(x \geq 0 \rightarrow \langle \mathrm{x} := \mathrm{x} + 1\rangle(x > 0).$$

De universele kwantor bindt hier de voorkomens van x in $x \geq 0$ en in $\mathrm{x} + 1$, de toekenning bindt de x in $x > 0$. Dat dit zo kunnen we inzien door herbenoemen van gebonden variabelen. Formule (0) is immers equivalent met:

$$\forall x(x \geq 0 \rightarrow \langle \mathrm{y} := \mathrm{x} + 1\rangle(y > 0).$$

Merk echter op dat "dubbel gebruik" van variabelen niet te elimineren is wanneer de ophoging $\mathrm{x} := \mathrm{x} + 1$ plaats vindt binnen een **zolang**-opdracht.

Opeenvolging van opdrachten wordt uiteraard als volgt weergeven:

Axioma 18.4 (opeenvolging)

$$\vdash [\textbf{begin } \pi_1;\cdots;\pi_n \textbf{ einde}]\varphi \leftrightarrow [\pi_1]\cdots[\pi_n]\varphi.$$

Opgave 18.5 Laat zien dat dit opeenvolgingsprincipe semantisch geldig is.

De **als-dan-(anders**) opdrachten worden beschreven door de volgende axioma's, die onmiddellijk uit de semantische clausules zijn af te lezen:

Axioma 18.5 (als–dan)

$$\vdash [(\textbf{als } \psi \textbf{ dan } \alpha)]\varphi \leftrightarrow ((\psi \rightarrow [\alpha]\varphi) \wedge (\neg\psi \rightarrow \varphi)).$$

Axioma 18.6 (als–dan–anders)

$$\vdash [(\textbf{als } \psi \textbf{ dan } \alpha \textbf{ anders } \beta)]\varphi \leftrightarrow ((\psi \rightarrow [\alpha]\varphi) \wedge (\neg\psi \rightarrow [\beta]\varphi)).$$

Voor **zolang**-opdrachten ligt de zaak minder eenvoudig. Weliswaar is het niet zo moeilijk een geldig postulaat te verzinnen, maar daarmee is de kous nog niet af.

Axioma 18.7 (zolang-iteratie)

$$\vdash [\textbf{zolang } \psi \textbf{ doe } \alpha]\varphi \leftrightarrow ((\psi \rightarrow [\alpha][\textbf{ zolang } \psi \textbf{ doe } \alpha]\varphi) \wedge (\neg\psi \rightarrow \varphi)).$$

Opgave 18.6 Laat zien dat het bovenstaande postulaat ook als volgt geformuleerd kan worden:

$$[\textbf{zolang } \psi \textbf{ doe } \alpha]\varphi \leftrightarrow [(\textbf{als } \psi \textbf{ dan begin } \alpha; \textbf{ zolang } \psi \textbf{ doe } \alpha \textbf{ einde})]\varphi.$$

Het recursieve effect wordt door dit schema wel verantwoord, maar op deze manier zullen we allerlei andere waarheden zoals een formule met slechts één voorkomen van een **zolang**-opdracht vrijwel nooit kunnen afleiden. We herinneren daarom aan een observatie die eigenlijk al in voorbeeld 18.4 gedaan is:

Axioma 18.8 (deconditionering)

$$\vdash [\textbf{zolang } \psi \textbf{ doe } \alpha]\neg\psi.$$

Onmiddellijk nadat een programma een **zolang**-opdracht gepasseerd is zal de conditie uit die opdracht niet (meer) gelden. Daarnaast willen we ook de beschikking hebben over formules die hun waarheid juist behouden bij het passeren van een **zolang**-opdracht:

Afleidingsregel 18.2 (conservatie)

$$\frac{\vdash \quad \varphi \rightarrow (\psi \rightarrow [\alpha]\varphi)}{\vdash \quad \varphi \rightarrow [\textbf{zolang } \psi \textbf{ doe } \alpha]\varphi.}$$

Om te zien hoe we redeneren in een dergelijk modaal systeem zullen we enige stellingen en regels afleiden. Daarna besteden we in het bijzonder aandacht aan de Hoare-correctheidsregels, die immers ook binnen het ruimere kader van de dynamische logica moeten gelden.

We bewijzen eerst een eenvoudige maar nuttige regel die zelfs in normale modale logica afleidbaar is:

Stelling 18.1

$$\vdash \varphi \rightarrow \psi \Rightarrow \vdash [\pi]\varphi \rightarrow [\pi]\psi.$$

Bewijs:

1	$\varphi \rightarrow \psi$	[gegeven stelling]
2	$[\pi](\varphi \rightarrow \psi)$	[uit 1 met regel N]
3	$[\pi](\varphi \rightarrow \psi) \rightarrow ([\pi]\varphi \rightarrow [\pi]\psi)$	[axioma K]
4	$[\pi]\varphi \rightarrow [\pi]\psi.$	[met MP uit 2 en 3]

□

Zonder bewijs vermelden we ook de volgende bekende eigenschap van normale modale logica's:

Stelling 18.2

$$\vdash [\pi](\varphi \wedge \psi) \leftrightarrow ([\pi]\varphi \wedge [\pi]\psi).$$

Een resultaat dat de specifieke eigenschappen van de dynamische axiomatiek illustreert is een versterking van het toekenningsaxioma:

Stelling 18.3

$$\vdash [v := t]\varphi \leftrightarrow [t/v]\varphi.$$

Bewijs:

1	$[v := t]\varphi \rightarrow [t/v]\varphi$	[toekenningsaxioma]
2	$[v := t]\neg\varphi \rightarrow [t/v]\neg\varphi$	[toekenningsaxioma]
3	$[v := t]\neg\varphi \rightarrow \neg[t/v]\varphi$	[uit 2 met substitutie]
4	$[t/v]\varphi \rightarrow \neg[v := t]\neg\varphi$	[met contrapositie uit 3]
5	$[t/v]\varphi \rightarrow \langle v := t\rangle\varphi$	[met definitie $\langle\pi\rangle$ uit 4]
6	$\langle v := t\rangle\varphi \rightarrow [v := t]\varphi$	[determinisme]
7	$[t/v]\varphi \rightarrow [v := t]\varphi$	[met propositielogica uit 5 en 6]
8	$[v := t]\varphi \leftrightarrow [t/v]\varphi.$	[met propositielogica uit 1 en 7]

□

Enigszins paradoxaal is stelling 18.3 in het licht van de eerdere opmerking dat de toekenningopdracht variabelen bindt: aan de rechterkant van de equivalentie bij de stelling hoeft helemaal geen sprake te zijn van een gebonden variabele. Enige reflectie leert dat hier geen werkelijke tegenspraak bestaat.

Opgave 18.7 Verklaar de bovenstaande schijnbare paradox.

Toegerust met het bovenstaande kunnen we een resultaat van groter belang voor de dynamische logica aantonen:

Stelling 18.4 *De axiomatiek van Floyd–Hoare is afleidbaar binnen de DL.*

Bewijs: We zullen niet alles uitputtend bewijzen, maar volstaan soms met enige opmerkingen en laten de rest aan de lezer.

– Het *toekenningsaxioma* uit de correctheidslogica krijgt dynamisch de vorm $[t/v]\varphi \to [v := t]\varphi$ en dat is gewoon één kant van de eerder in stelling 18.3 bewezen versterking van het dynamische toekenningsaxioma.

– *Versterking van beginvoorwaarde* is nu zelfs een propositioneel geldige gevolgtrekking.

– *Afzwakking van eindvoorwaarde* leiden we wel even af:

1	$\varphi \to [\pi]\psi$	[gegeven stelling]
2	$\psi \to \chi$	[gegeven stelling]
3	$[\pi]\psi \to [\pi]\chi$	[uit 2 met stelling 18.1]
4	$\varphi \to [\pi]\chi$	[uit 1 en 3 met propositielogica].

– De *opeenvolgingsregel* laten we over aan de lezer.

– De *als–dan regel* en de *als–dan–anders regel* laten zich ook tamelijk rechtstreeks afleiden.

– De *zolang regel* heeft iets meer voeten in de aarde:

1	$(\varphi \wedge \sigma) \to [\pi]\varphi$	[gegeven stelling]
2	$\varphi \to (\sigma \to [\pi]\varphi)$	[uit 1 met propositielogica]
3	$\varphi \to [\textbf{zolang } \sigma \textbf{ doe } \pi]\varphi$	[uit 2 met conservatie]
4	$[\textbf{zolang } \sigma \textbf{ doe } \pi]\neg\sigma$	[deconditionering]
5	$\varphi \to [\textbf{zolang } \sigma \textbf{ doe } \pi]\neg\sigma$	[uit 4 met pL]
6	$\varphi \to ([\textbf{zolang } \sigma \textbf{ doe } \pi]\varphi \wedge$ $[\textbf{zolang } \sigma \textbf{ doe } \pi]\neg\sigma)$	[uit 5 en 6 met pL]
7	$\varphi \to [\textbf{zolang } \sigma \textbf{ doe } \pi](\varphi \wedge \neg\sigma)$	[uit 6 met stelling 18.2].

□

Opgave 18.8 Leid de dynamische weergave van de Floyd–Hoare opeenvolgingsregel af; dat wil zeggen: bewijs de volgende regel

$$\frac{\vdash \varphi \to [\pi_1]\psi \qquad \vdash \psi \to [\pi_2]\chi}{\vdash \varphi \to [\textbf{begin } \pi_1;\pi_2 \textbf{ einde}]\chi.}$$

Over dit axiomastelsel valt nog veel meer te zeggen maar we laten het hierbij. Voor verwante resultaten, ook ten aanzien van een krachtiger (zogenaamd reguliere) programmeertaal verwijzen we wederom naar [Harel 1984]. Een aantal op zichzelf schijnbaar eenvoudige noties zoals iteratie en terminatie blijken dan overigens anders geïnterpreteerd te moeten worden; zie voor deze subtiliteiten ook [Harel 1979].

Floyd–Hoare logica en dynamische logica vormen interessante toepassingen van logische technieken op het gebied van het programmeren. Hoewel de meeste programmeurs de tijd of de formele achtergrond missen om dit heilzame logische middel te gebruiken, zijn wijzelf ten volle overtuigd van het nut van deze toepassingen.

Afgezien van het onmiskenbare belang van de dynamische logica voor de informatica, is de dynamische logica de laatste jaren een belangrijke inspiratiebron geworden voor onderzoekers in de semantiek van natuurlijke talen. Sinds het begin van de jaren 80 zijn er al semantische theorieën opgesteld die uitgaan van een "van links naar rechts" aangroeien van de betekenis van een zin of tekst; zie bij voorbeeld [Kamp 1981] en [Heim 1982]. Het verband met DL wordt echter duidelijker gelegd in [Barwise 1987] en, nog explicieter, in [Groenendijk & Stokhof 1987]. Net zoals een werkend programma een verandering in de machinetoestand teweegbrengt, zorgt de uiting van een zin voor een verandering in de informatie waarover de toehoorder beschikt. Stokhof en Groenendijk trekken de analogie met de semantiek van DL nog verder door: de informatietoestanden—althans de onderdelen daarvan waarop zij zich concentreren—zijn weer *bedelingen*. Voor de taalkundige semantiek zijn bedelingen van groot belang omdat een aantal hardnekkige problemen nu juist te maken hebben met het binden van variabelen. Meer taalkundig: hoe kunnen in bepaalde gevallen pronomina die buiten het bereik van een kwantificerende uitdrukking vallen daar toch door gebonden worden? Beschouw de volgende voorbeelden:

Er loopt een man op straat. Hij fluit. (1)

Iedere boer die een ezel heeft slaat hem. (2)

[Groenendijk & Stokhof 1987] merken op dat we de anaforische lezingen van (1) en (2) toch eenvoudig in predikatenlogica kunnen weergeven:

$$\exists x(Mx \wedge Lx \wedge Fx) \tag{3}$$

$$\forall x \forall y((Bx \wedge Ey \wedge Hxy) \to Sxy). \tag{4}$$

Hiermee is de directe correspondentie tussen de opbouw van de tekst respectievelijk de zin enerzijds en de logische representatie anderzijds echter om zeep geholpen. Dat is op zichzelf nog geen ramp, zou je kunnen denken: in § 11.4 hebben we gezien dat bij voorbeeld Russell's analyse van het bepaald lidwoord ook de zinsstructuur niet volgt. Er is echter een verschil: Russell's descriptie-theorie kan namelijk ook worden weergegeven in typenlogica, en dan bestaat er weer wel een directe correspondentie tussen syntactische structuur en logische representatie. Zo niet

bij (3) en (4). In jargon: deze formules zijn niet *compositioneel* verkregen. Een vervelend gevolg daarvan is dat het niet meer duidelijk is hoe we de betekenisrepresentaties automatisch uit de zinnen kunnen verkrijgen. Nu kan men wel een niet-compositionele procedure definiëren die toch bij voorbeeld (4) uit (2) afleidt. Zo'n procedure kan als volgt luiden:

1. Vertaal de zin compositioneel in een formule waarin de anafoor door een vrije variabele, zeg x, wordt weergegeven.

2. Breng de formule in zogenaamde prenex normaalvorm, dat wil zeggen: geef een equivalente formule waarbij alle kwantoren voorop staan.

3. Vervang x door een willekeurige andere variabele.

Het zal echter lastig zijn een dergelijke procedure zo in te perken dat zij niet teveel lezingen genereert. Hoe dan ook, [Groenendijk & Stokhof 1987] komen met de volgende elegante oplossing: stel dat we de zinnen toch weergeven door een compositioneel te verkrijgen representatie, namelijk voor bovenstaande voorbeelden:

$$\exists x(Mx \wedge Lx) \wedge Fx \tag{5}$$

$$\forall x((Bx \wedge \exists y(Ey \wedge Hxy)) \rightarrow Sxy). \tag{6}$$

Op het eerste gezicht lijken deze formules de verkeerde betekenissen weer te geven van (1) en (2), namelijk de weinig voor de hand liggende lezingen waarin het pronomen (*hij*, *hem*) vrij (deiktisch) is. Maar nu komt de aap uit de mouw: deze doodgewone formules krijgen een *dynamische* interpretatie, als volgt:

– Niet alleen opdrachten, maar ook logische formules veranderen de waarden van variabelen. In essentie worden conjunctie en concatenatie van zinnen in de semantiek als opeenvolgingsopdrachten geïnterpreteerd.
– Existentiële kwantificatie wordt opgevat als een speciale vorm van een toekenningsopdracht.

De dynamische kijk op de existentiële kwantor wordt aannemelijk als je denkt aan de overeenkomsten tussen het 'bindend effect' van de logische kwantoren en dat van de (modaal geïnterpreteerde) toekenningsopdrachten.

Het gezamenlijke effect van de twee dynamiserings-stappen is dat existentiële kwantoren bedelingen 'aanpassen', en dat op zo'n manier dat die aangepaste bedelingen daarna bewaard blijven voor de interpretatie van vrije variabelen buiten het bereik van de kwantor (in onze voorbeelden: voor de interpretatie van de vrije variabele die de anafoor vertegenwoordigt). Voor verder details verwijzen we naar [Groenendijk & Stokhof 1987]. Het is buiten kijf dat er aan deze toepassing van dynamische logica op de semantiek van natuurlijke taal nog druk moet worden gesleuteld om de empirische feiten te verantwoorden. Dat neemt echter niet weg dat we hier een aardige demonstratie hebben van de bruikbaarheid van de 'dynamische kijk' op informatie, en bovendien een illustratie van de manier waarop logica, taalkunde en informatica elkaar positief kunnen beïnvloeden.

Het volgende is meer een afspraak met jezelf dan met ons. We verplichten je tot niets.

Afspraak

Altijd zal ik opgaven eerst zelf proberen, voordat ik bij de antwoorden kijk.

Als ik meteen bij de antwoorden kijk ben ik een slappeling die niets probeert, en daardoor niets leert.

Door eerst opgaven zelf te proberen word ik sterk. Ik leer door te proberen en daardoor onthoud ik meer.

Bijlage A

Antwoorden

Ons tekstverwerkingsprogramma LaTeX refereert naar items binnen opgaven met alleen het itemnummer, zonder dat dit voorafgegaan wordt door het eigenlijke opgavenummer. Op deze manier zijn antwoorden soms lastig te vinden. Helaas is het ons niet gelukt dit te repareren.

Bij het zoeken naar antwoorden raden wij aan eerst het juiste bladzijdenummer te vinden, en dan het juiste itemnummer.

Blz. 5, it. 1.1: Uitspraken in even rijen en in de kolom uiterst rechts zijn waar. De andere uitspraken zijn onwaar.

Blz. 5, it. 1.2: Dit is geen goed idee om twee redenen. Ten eerste steunen deze definities op intuïtie. Intuïtief weet iemand hoe de reeks $1,2,3,\ldots$ moet worden voortgezet, en is er weinig gevaar voor misverstand. Maar definities mogen niet zijn gebaseerd op intuïtie en een aanname van een gemeenschappelijk getalsbegrip. Ten tweede is er dan nog het probleem hoe we $\mathbb{Q}$, of nog erger, $\mathbb{R}$ zouden moeten noteren.

Blz. 5, it. 1.3: Normaal geldt $\mathbb{Z}^+ = \mathbb{N}$ en heeft het gebruik van een extra symbool dus niet zo veel nut. In de berekenbaarheidstheorie behoort 0 tot $\mathbb{N}$ en geldt dus dat $\mathbb{Z}^+ \neq \mathbb{N}$.

Blz. 6, it. 1.4: Te bewijzen $|x| \geq 0$ voor alle $x \in \mathbb{R}$. Zij $x \in \mathbb{R}$ willekeurig. Er zijn twee mogelijkheden: $x \geq 0$ of $x < 0$.

1. Als $x \geq 0$, dan per definitie $|x| = x$. Dus ook $|x| \geq 0$.
2. Als $x < 0$, dan per definitie $|x| = -x$. Dus ook $-|x| = x$. Dus $-|x| < 0$. Dus $|x| > 0$. Dus zeker $|x| \geq 0$. Dat laatste stapje is zeer belangrijk. Zie je wat daar gebeurd?

Het gevraagde is nu bewezen voor alle mogelijke waarden $x \in \mathbb{R}$.

Blz. 7, it. 1.5: We moeten laten zien dat $-x + y \geq x + y$, als we weten dat $x < 0$, $y > 0$ en $x + y > 0$. Welnu, aan beide kanten y aftrekken en x optellen en tenslotte delen door twee geeft $0 \geq x$, wat waar is.

Dus de ongelijkheid $-x + y \geq x + y$ klopt, als we weten dat $x < 0$, $y > 0$ en $x + y > 0$. Dus de ongelijkheid $|x| + |y| \geq |x + y|$ klopt in geval $x < 0$, $y > 0$ en $x + y > 0$.

Blz. 7, it. 1: Te bewijzen:

$$-v \leq u \leq v \text{ als en alleen als } |u| \leq v. \tag{1}$$

Merk eerst op dat voor zowel de linker- als rechterkant v niet negatief kan zijn: voor de linkerkant niet, omdat u dan tussen een positief en negatief getal in zou zitten wat (in die volgorde) onmogelijk is, en aan de rechterkant omdat absolute waardes nooit negatief zijn.

We onderscheiden twee gevallen: $u \geq 0$ en $u < 0$. Als $u \geq 0$ dan $|u| = u$ en staat aan de rechterkant $u \leq v$. Merk op dat de linker- en rechterkant daarmee beiden waar of onwaar zijn.

Als $u < 0$ geldt $|u| = -u$ en staat aan de rechterkant $-u \leq v$, oftewel $-v \leq u$. Weer geldt dat de linker- en rechterkant daarmee beiden waar of onwaar zijn.

Merk op dat beide gevallen gebruik maken van het feit dat v niet negatief is.

Blz. 7, it. 2: Zij x en y willekeurig. Er geldt uiteraard $|x| \leq |x|$. Vanwege (1) geldt dus

$$-|x| \leq x \leq |x|. \tag{2}$$

Evenzo

$$-|y| \leq y \leq |y|. \tag{3}$$

Vergelijkingen (2) en (3) optellen geeft

$$-|x| - |y| \leq x + y \leq |x| + |y|. \tag{4}$$

Identiteit (1) terug toepassen op (4) geeft

$$|x + y| \leq ||x| + |y||. \tag{5}$$

Omdat $|x| + |y| \geq 0$ geldt $||x| + |y|| = |x| + |y|$. Dus

$$|x + y| \leq |x| + |y|. \tag{6}$$

Blz. 7, it. 1: Ledigerwijs waar.

Blz. 7, it. 2: Ledigerwijs waar.

Blz. 7, it. 3: $A = \{\pm\sqrt{2}\}$ en niet waar

Blz. 7, it. 4: Ledigerwijs waar

Blz. 7, it. 5: Niet waar. (Neem bijvoorbeeld $(x, y, z) = (6, 8, 10)$.)

Blz. 7, it. 6: Ledigerwijs waar.

Blz. 7, it. 7: Volgens de laatste stelling van Fermat zijn er alleen maar triviale oplossingen. Voor deze oplossingen geldt $|x|, |y|, |z| \leq 0$. Makkelijk is in te zien dat de linkerkant $x + y$ van $x + y \leq z + 3$ hiermee nooit groter dan de rechterkant $z + 3$ kan worden. De bewering is waar, maar niet ledigerwijs waar.

Blz. 8, it. 1.8: $-15+24=9$, $-17+2=-15$, $-20+14=-6$, $-5+17=12$. Dus koppen verdwijnen en verschijnen in veelvouden van drie. Maar 100 (het vertrekpunt) is geen veelvoud van 3 en 0 (het doel) is dat wel. Dus kan dit proces nooit op nul uitkomen.

Nu voor de invariant. Het getal 100 behoort tot de familie getallen die bij deling door 3 rest 1 opleveren. Dit is de familie

$$\ldots,-5,-2,1,4,7,11,\ldots,100,\ldots$$

We schrijven:

$$100 \equiv 1 \pmod 3.$$

Laat k het aantal koppen zijn. Dan geldt aan het begin, als $k=100$, dat $k \equiv 1 \pmod 3$. Veelvouden van 3 bij k optellen of aftrekken veranderd daar niets aan. We blijven binnen dezelfde familie. Dus de invariant is hier $k \equiv 1 \pmod 3$. Het getal $k=0$ kan nooit worden bereikt want $k \equiv 0 \pmod 3$.

Blz. 8, it. 1.9: Allereerst merken we op dat bij elke trekking er twee bonen verwijderd worden, en één teruggelegd. Na 49 trekkingen zal er dus nog één boon over zijn. Het ondoenlijk alle mogelijkheden te proberen, dat zijn er heel erg veel. In plaats daarvan zoeken we naar een geschikte invariant. We kijken naar het netto effect op het aantal witte en zwarte bonen bij elk van de vier mogelijke trekkingen:

trekking	actie	Z	W
$-ZZ$	$+Z$	-1	0
$-WW$	$+Z$	$+1$	-2
$-ZW$	$+W$	-1	0
$-WZ$	$+W$	-1	0

Het aantal witte bonen in de kan blijft gelijk of neemt met twee af. Het oneven-zijn van het aantal witte bonen is dus een invariant in dit spel. Het aantal witte bonen blijft dus altijd oneven, in het bijzonder aan het eind. De laatste boon is dus altijd wit.

Blz. 8, it. 1.10:

TREKKING TERUG
Dezelfde kleur ⇒ één van de drie.
Verschillende kleur ⇒ rood.
Twee dezelfde ⇒ afwijkende terug + blauwe.

TREKKING	R	W	B	WEG
RRR	-2	0	0	-2
WWW	0	-2	0	-2
BBB	0	0	-2	-2
WBR	0	-1	-1	-2
$RR*$	-2	0	$+1$	-1
$WW*$	0	-2	$+1$	-1
$BB*$	0	0	-1	-1

Mono-variant: aantal ballen in de doos. Die daalt met elke trekking.
Invariant: #rode ballen = 0 (mod 2)

De clou in genoemde opgave zit in het kijken naar het aantal rode ballen. Zoals je aan de tabel kunt zien blijft dat aantal bij elke stap gelijk of wordt twee minder. Aanvankelijk is het 100. Dus een invariant is dat het aantal rode ballen even is. Dus kan een situatie met precies een blauwe en een rode bal niet bereikt worden, want dan is het aantal rode ballen niet even.

Eigenlijk hetzelfde argument als bij de bonen in de koffiekan (in het dictaat Beschrijven en Bewijzen). Deze opgave werd destijds bedacht als een iets ingewikkelder variant daarop met drie kleuren. Er is ook nog wat ruis toegevoegd: in speciale gevallen wordt er nog een extra blauwe teruggelegd, maar dat doet er allemaal niet toe.

Blz. 8, it. 1.11: Er wordt begonnen met een axioma, `MI`, waarvan het aantal `I`'s geen drievoud is, dat wil zeggen, gelijk is aan 1 of 2 (mod 3). Bewering:

> Het aantal `I`'s in elke geproduceerde string
> kan nooit een drievoud worden.

De bewering is dus dat

> aantal `I`'s(x) $\neq 0 \pmod 3$

een invariant is op de verzameling van geproduceerde strings (x is hier een string). Als deze bewering waar is, dan kan het verlangde resultaat, `MU`, nooit bereikt worden, want het aantal `I`'s in deze string is namelijk WEL een drievoud (namelijk: 0).

Bewijs van de bewering

1. "Achter een rijtje met een `I` aan het eind mag je een `U` zetten."

   Het aantal `I`'s veranderd niet, en blijft dus geen drievoud.

2. "Van het rijtje `Mx` mag je `Mxx` maken, voor elk rijtje symbolen `x`."

   Het aantal `I`'s wordt tweemaal zo groot. Het aantal `I`'s was geen drievoud, en blijft geen drievoud:

   $$k=1 \pmod 3 \quad\Rightarrow\quad 2k=2 \pmod 3,$$
   $$k=2 \pmod 3 \quad\Rightarrow\quad 2k=4=1 \pmod 3.$$

3. "Als er `III` in een rijtje staat, mag je dat vervangen door `U`."

   Er verdwijnen drie `I`'s. Duidelijk is dat, als het aantal `I`'s in de originele string geen drievoud is, het aantal `I`'s in de resulterende string dat ook niet is:

   $$k \neq 0 \pmod 3 \quad\Rightarrow\quad k-3 \neq 0 \pmod 3.$$

4. "Als er `UU` in een rijtje staat, mag je dat weglaten."

   Het aantal `I`'s veranderd niet.

Blz. 9, it. 1.12: Op een bord van 2×2 punten kan A niet winnen: er zijn minstens 4 lijnen nodig om een gesloten figuur te maken. Laten we daarom eerst kijken of er een winnende strategie voor B is. Het spel vaak spelen geeft weinig inzicht in een strategie. Daarom kijken we naar de eigenschappen van gesloten figuren. Als B kan

verhinderen dat één van deze eigenschappen op het bord komt, kan ze winnen. De invariant luidt dan: het bord bevat geen figuur met de betreffende eigenschap.

Welke eigenschappen heeft een gesloten figuur? Er zitten parallelle lijnen in, maar die kan B niet verhinderen. Er zit een even aantal parallelle lijnen in, maar ook dat kan B niet verhinderen. Er zitten tenminste vier hoeken in, maar B kan niet verhinderen dat A hoeken tekent. Er zit minstens één L-vormige hoek in met de punt links onder, en dé kan B verhinderen! Elke keer als A een horizontale of verticale lijn trekt, kan B deze aanvullen tot een L-vorm (tenzij A op de boven- of rechterrand zet; dan mag B een willekeurige zet doen). A kan dan nooit zelf een L-vorm in zijn figuur maken, en dus geen gesloten figuur. Op grond van de invariant: "het bord bevat geen L-vormen van A" kunnen we concluderen: B heeft een winnende strategie.

Blz. 9, it. 1: Bekijk de totale bebouwingsomtrek. Snel is in te zien dat als een tegel twee buren heeft en bebouwd raakt, de totale bebouwingsomtrek niet toe- of afneemt. Immers, twee bestaande grenskanten worden dichtgemetseld, en het nieuwe blok voegt twee nieuwe grenskanten toe. Hoe de bebouwde buren liggen ten opzichte van de centrale patch maakt daarbij niet uit. (Ga na!)

Ook is snel in te zien dat als een tegel bebouwde drie buren heeft en bebouwd raakt, de totale bebouwingsomtrek afneemt met twee. Bij een tegel met vier bebouwde buren neemt de totale bebouwingsomtrek zelfs af met vier. (Ga na!) *De totale bebouwingsomtrek neemt dus altijd af.*

Blz. 9, it. 2: Stel het veld bezit afmetingen $m \times n$. De omtrek van een volledig bebouwd veld is dus $m+m+n+n = 2(m+n)$. Bij een begin-bebouwingsomtrek kleiner dan $2(m+n)$ kan een veld dus nooit vol raken, immers de bebouwingsomtrek daalt altijd. Dus

$$2(m+n)-1$$

is een maximale begin-bebouwingsomtrek. Zo'n omtrek kan worden bereikt als alle bebouwde tegels in het begin geïsoleerd ("los") liggen. Omdat de begin-bebouwing random wordt gegenereerd, kan dit laatstgenoemde scenario niet worden uitgesloten. De maximale begin-bebouwingsomtrek moet dus door vier gedeeld worden. Dat is het maximaal aantal tegels. Omdat delen door vier meestal een gebroken getal oplevert, wordt het antwoord

$$N = \lfloor \frac{2(m+n)-1}{4} \rfloor,$$

waarbij "$\lfloor\cdot\rfloor$" de floor-functie is. (I.e., $\lfloor 4.3 \rfloor = 4$, $\lfloor 5.9 \rfloor = 5$, etc.) Bij vierkante velden wordt dit natuurlijk

$$N = \lfloor \frac{4n-1}{4} \rfloor.$$

Een tabelletje:

Zijde:	2	5	10	100	1000
Max. begin-bebouwing:	1	4	9	99	999

Blz. 9, it. 3: Bij een 10×10 veldje met 9 bebouwde tegels is de begin-bebouwingsomtrek maximaal $4 \times 9 = 36$. (Dat is als alle tegels los liggen.) Met 9 bebouwde tegels kan de bebouwing dus nooit de omtrek van het veldje aannemen (40).

Blz. 9, it. 4: Leg alle tegels op een diagonaal. Het is nu makkelijk in te zien dat deze diagonaal blijft groeien. (Ga na!)

Blz. 9, it. 5: De gevonden bovengrens is maximaal. Hiertoe laten we zien dat

$$\lceil \frac{2(m+n)}{4} \rceil = \lceil \frac{m+n}{2} \rceil$$

tegels voldoende zijn om een veldje te laten vollopen. Hier is "$\lceil\cdot\rceil$" de ceiling-functie. (I.e., $\lceil 4.3 \rceil = 5$, $\lceil 5.9 \rceil = 6$, etc.)

Voor vierkanten is het makkelijk: legt bij een $n \times n$ vierkant een diagonaal van n tegels. Deze diagonaal laat vervolgens het hele veld dichtgroeien. (Ga na!)

Voor rechthoeken werkt de diagonaalmethode niet. We zouden het diagonaal-idee kunnen omzetten naar een soort zig-zag, maar ook dit werkt niet.

Wat wel effectief blijkt om een rechthoek vol te laten lopen is om een soort *kantelen* (stippellijn) aan te leggen: leg zowel aan de linker- als aan de onderkant om en om een tegel. Als tenminste één zijde een even lengte bezit, leg dan nog één tegel in de rechterbovenhoek. Gemakkelijk is na te gaan dat in alle gevallen de rechthoek nu volloopt.

We moeten nog laten zien dat in elk van de gevallen het gebruikte aantal tegels inderdaad nooit de $\lceil (m+n)/2 \rceil$ overschrijdt.

- Als m en n beide even zijn, komen we uit op

  $$\frac{m}{2} + \frac{n}{2} - 1 + 1 = \frac{m+n}{2}$$

  tegels: $m/2$ voor één kant, $n/2-1$ voor de andere kant (min 1, want de hoek is al bezet) plus nog één voor de rechter-bovenhoek.

- Als m en n beide oneven zijn, komen we uit op

  $$\frac{m+1}{2} + \frac{n+1}{2} - 1 = \frac{m+n}{2}$$

  tegels: $(m+1)/2$ voor één kant, $(n+1)/2-1$ voor de andere kant (niet +1, want de hoek loopt zonder extra tegel al vol). Een tegel voor de rechter-bovenhoek is niet nodig.

- Als m even is en n oneven, komen we uit op

  $$\frac{m}{2} + \frac{n+1}{2} - 1 + 1 = \frac{m+n}{2} + \frac{1}{2} \qquad (7)$$

  tegels. In dit geval is het nu weer nodig een tegel in de rechter-bovenhoek te leggen. (Ga na!) Omdat m even is en n oneven, eindigt $(m+n)/2$ op een half, en is (7) geheel. In het bijzonder geldt

  $$\frac{m+n}{2} + \frac{1}{2} \le \lceil \frac{m+n}{2} \rceil,$$

  hetgeen bewezen moest worden.

De bewijstechniek kan "een bewijs met gebruikmaking van een invariant" worden genoemd. De invariant is hier:

> bebouwingsomtrek ≤ bebouwingsomtrek van de start-configuratie.

Als de tegelvloer groeit is de bebouwingsomtrek geen mono-variant: een mono-variant moet altijd stijgen en dat is hier niet altijd het geval.

Blz. 10, it. 1.14: First, we separate the members arbitrarily into two factions. Let H be the sum of all the enemies each member has in his own faction. Suppose one member (letŠs call him Bob) has at least two enemies in his own faction. Then if Bob switches factions, H will decrease. Let this process continue for all members in the same situation as Bob. Since H is a positive integral function that decreases at each step of the process, the process must terminate. At this point, no senator can have more than one enemy in his own faction, because otherwise the process (by definition) would not have terminated. Thus we have found the desired division of senators.

Alternative proof, based on the extremal principle: consider all partitions of the Senate into factions, and count the total number of enemies E that senators have in their factions. Now pick a partition of the Senate with minimal E. This partition has the desired property: if some senator had more than one enemy within his faction, we could move him to the other faction and decrease E, which would be a contradiction.

Blz. 10, it. 1: Dit is een implicatie. Door de bank genomen geldt deze implicatie, tenzij we flauw gaan doen over vrouwen met baarden, etc. (Wat kan. Zulke vrouwen werden vroeger op rondreizende kermissen tentoongesteld. Je zou ook kunnen Googelen op "woman with beard," maar dat gaat al aardig richting soggen.) Laten we even aannemen dat de implicatie geldt. Contrapositie:

> "Alle niet-mannen dragen geen baard."

Seksuele pathologieën daargelaten kan worden aangenomen dat alle mensen die geen man zijn, vrouw zijn. Dan is de contrapositie:

> "Alle vrouwen dragen geen baard."

Ook hier geldt weer dat de bewering waar is als het bestaan van baarddragende vrouwen wordt uitgesloten. Als het bestaan van baarddragende vrouwen *niet* wordt uitgesloten zijn zowel de implicatie als de contra-implicatie onwaar.

Blz. 10, it. 2: Universum: alle mensen. (Of: alle *levende* mensen. Of: alle reizigers.) Dit is een implicatie. De implicatie lijkt waar: als je een auto hebt, heb je per definitie vervoer, pathologische gevallen uitgesloten, zoals kapotte auto's, auto's zonder benzine, etc. Contrapositie:

> "Alle mensen zonder vervoer rijden geen auto."

(Of: alle levende mensen zonder vervoer rijden geen auto. Of: alle reizigers zonder vervoer rijden geen auto.) Dit is dan ook waar, ook weer onder het voorbehoud dat pathologische gevallen worden uitgesloten.

Blz. 10, it. 3: Universum: alle kinderen. (Of: alle schoolgaande kinderen. Of: alle in Nederland geboren kinderen.) Dit is een implicatie. Contrapositie:

> "Alle kinderen die niet op de basisschool zitten komen niet uit groep 5."

Blz. 10, it. 4: Universum: alle sporters. Dit is een implicatie. Contrapositie:

> "Als een sporter mag meedoen, dan is deze niet gediskwalificeerd."

Je kunt de contrapositie realistisch maken door zoiets te zeggen als:

> "Als een sporter mag meedoen, dan is deze blijkbaar niet gediskwalificeerd geworden."

Alle drie de uitspraken (implicatie, contrapositie en variatie op contrapositie) komen op hetzelfde neer.

Blz. 10, it. 5: Universum: alle mensen. (Of: alle kiesgerechtigden.) Dit is een implicatie. Deze implicatie is waar: een stem kan maar één keer worden uitgebracht. Contrapositie:

> "Kun je nog op partij B stemmen, dan heb je (blijkbaar) (nog) niet op partij A gestemd."

Ook de contrapositie is waar: als het nog mogelijk is om op een partij, bijv. B, te stemmen, dan heb je je stem blijkbaar nog niet uitg=ebracht, in het bijzonder niet op A.

Blz. 10, it. 6: Universum als in vorige onderdeel. Dit is een implicatie. Deze implicatie is onwaar: je kan op een andere partij $C \neq A$ hebben gestemd en zo je stem al hebben weggegeven. Contrapositie:

> "Kun je niet meer op partij B stemmen, dan heb je (blijkbaar) op partij A gestemd."

Ook de contrapositie is onwaar: je kunt op een andere partij hebben gestemd, of je kunt om wat voor reden dan ook niet in staat zijn om te stemmen.

Blz. 10, it. 7: Voorbeeld van een universum: alle vrouwen waarbij een zwangerschapstest wordt afgenomen. Dit is een implicatie. Als "false positives" (onterechte signaleringen) buiten beschouwing worden gelaten, dan is deze implicatie waar. Contrapositie:

> "Als je niet in verwachting bent, dan moet de je zwangerschapstest negatief uitvallen."

Ook de contra-implicatie is waar als valse positieven buiten beschouwing worden gelaten.

Blz. 10, it. 8: Universum als in het vorige onderdeel. Dit is toegegeven een moeilijke opgave. De bewering

> "Als je zwangerschapstest negatief is, dan is er toch nog een mogelijkheid dat je in verwachting kunt zijn."

oppert de *mogelijkheid* dat een test vals negatief is, wat zo veel wil zeggen dat een zwangerschapstest kan falen in de aanwezigheid van een feitelijke zwangerschap. Dit is een reële mogelijkheid als er bijvoorbeeld te vroeg in de zwangerschap wordt getest, of als de vrouw in de periode voorafgaand aan de test erg veel heeft gedronken. Laten we deze bewering dus als waar aanmerken.

Deze bewering is *geen* implicatie, omdat de rechterkant spreekt over een *mogelijkheid*. Zodra dat gebeurd is er geen sprake meer van een categorische (i.e., altijd geldende) implicatie, maar van een op zichzelf staand voorbeeld. Een *uitzondering*, zo je wilt. Dezelfde bewering, maar dan anders geformuleerd:

> "Het is mogelijk dat een zwangerschapstest negatief uitvalt terwijl je toch in verwachting bent."

Weer anders geformuleerd, en zo dat de logisch structuur tot uiting komt:

> "Er is een situatie denkbaar waarin de volgende twee beweringen tegelijkertijd gelden:
>
> 1. Je bent in verwachting.
> 2. Je zwangerschapstest valt negatief uit."

In de logica wordt deze bewering gezien als een *conjunctie* (een samenvoeging van twee beweringen met behulp van het voegwoord "en"), die voorafgegaan wordt door een mogelijkheidsindicator. In de modale logica zou deze bewering worden opgeschreven als

$$\Diamond(p \wedge q),$$

waarbij "$\Diamond$" staat voor "het is mogelijk dat", "p" voor bewering (1), en "q" voor bewering (2). Zie verder ook de discussie in Hoofdstuk 6 op blz. 73.

Blz. 10, it. 1.16: Stel $n \in \mathbb{N}$ zodanig dat $\sqrt{n} \in \mathbb{Q}$. Dan $\sqrt{n} = k/l$ voor zekere $k, l \in \mathbb{N}$ met $l \neq 0$ en $\gcd(k,l) = 1$. Dus $k^2 = nl^2$. Intuïtief is nu al in te zien dat n een kwadraat moet zijn, want alle kwadraten die l^2 niet maakt aan de rechterkant, moeten door n gemaakt worden (immers er staat een kwadraat aan de linkerkant). We gaan dit netjes bewijzen. Stel $p \mid n$ is een priemdeler van n (p is dus priem). Omdat $k^2 = nl^2$ moet p nu ook in het linkerlid voorkomen, i.e., $p \mid k^2$. Omdat p priem is moet nu ook $p^2 \mid k^2$. (Immers, alle priemdelers van een kwadraat komen tenminste twee keer voor.) Maar dan moet p^2 ook het rechterlid delen, i.e., $p^2 \mid nl^2$. Dus $p^2 \mid n$ of $p^2 \mid l^2$. Het laatste, $p^2 \mid l^2$, zou impliceren dat $p \mid l$, maar dat kan niet, want eerder hadden we al $p \mid k$ en we hadden aangenomen dat $\gcd(k,l) = 1$. Dus moet $p^2 \mid n$.

We hebben zojuist bewezen dat voor iedere priemdeler $p \mid n$ geldt $p^2 \mid n$. Hieruit volgt dat n zelf een kwadraat moet zijn.

Misschien zijn er kortere bewijzen, dit is i.i.g. één bewijs.

Hoe *vind* je zo'n bewijs? Allereerst door je de contrapositie netjes uit te drukken. Eerst in woorden, en dan met variabelen. Vervolgens kom je terecht bij de uitdrukking $k^2 = nl^2$. Hier moet het inzicht danwel het vermoeden komen dat n, als deze omringd wordt door kwadraten, zelf ook een kwadraat moet zijn. Vervolgens ga je nadenken over hoe dit aangetoond kan worden. Dat kan door te beseffen dat kwadraten alle priemdelers dubbel hebben. (En alle getallen met dubbele priemdelers kwadraten zijn.) Vervolgens is het toch wel een kwestie van dóórredeneren, dat wil zeggen, alle consequenties van $k^2 = nl^2$ opschrijven totdat je bij de conclusie uitkomt dat elke priemdeler p het getal n twee keer moet delen.

Blz. 10, it. 1: Waar: als er nooit 5 of meer mensen in dezelfde maand jarig zijn, kan de groep uit ten hoogste $12 \times 4 = 48$ personen bestaan, en $48 < 50$.

Blz. 10, it. 2: Onwaar: het kan bijvoorbeeld voorkomen dat steeds 5 mensen in één van elke eerste tien maanden jarig zijn. Dus: $5,5,5,5,5,5,5,5,5,5,0,0$. Andere mogelijke verdelingen: $5,5,5,5,5,5,5,5,5,4,1,0$ of $5,5,5,4,5,5,5,5,5,4,1,1$ etc.

Blz. 11, it. 1.18: We bewijzen eerst

$$x \leq y \Rightarrow x^2 \leq y^2 \tag{8}$$

voor niet-negatieve getallen $x \leq y$, als volgt: als we de ongelijkheid $0 \leq x \leq y$ aan beide kanten met het niet-negatieve getal x vermenigvuldigen krijgen we

$$0 \leq x^2 \leq xy.$$

Als we diezelfde ongelijkheid $0 \leq x \leq y$ aan beide kanten met het niet-negatieve getal y vermenigvuldigen krijgen we

$$0 \leq xy \leq y^2.$$

Deze ongelijkheden achter elkaar zetten levert

$$0 \leq x^2 \leq xy \leq y^2$$

op, ergo $x^2 \leq y^2$.

We gaan nu de omgekeerde implicatie bewijzen, nl. $x^2 \leq y^2 \Rightarrow x \leq y$ voor niet-negatieve getallen $x \leq y$, als volgt: stel dat $x^2 \leq y^2$ voor niet-negatieve getallen $x \leq y$. Stel nu toch dat $x \leq y$ niet geldt. Dus $x \not\leq y$. Dus dan zou $y < x$ gelden. Dus zeker $y \leq x$. Via (8) concluderen we $y^2 \leq x^2$. Samen met onze aanname $x^2 \leq y^2$ levert dit $x^2 = y^2$. Twee kwadraten van positieve getallen zijn gelijk als en alleen de getallen zelf gelijk waren. Dus $x = y$. Maar dit is in tegenspraak met het eerder gevonden $y < x$. Dus onze aanname $x \not\leq y$ blijkt niet houdbaar. We kunnen niet anders dan concluderen dat $x \leq y$ moet gelden, waarmee het gevraagde bewezen is.

Blz. 11, it. 1.19: Stel n is even. Dus $n = 2m$. Dan is

$$n^2 = (2m)^2 = 4m^2 = 2\cdot 2m^2.$$

Dus n^2 is even.

Stel omgekeerd dat n oneven is. Dus $n = 2m+1$. Dan is

$$n^2 = (2m+1)^2 = 4m^2 + 4m + 1 = 2(2m^2 + 2m) + 1.$$

Dus n^2 is oneven.

Blz. 11, it. 1: Het bewijs is een kopie van het eerder gegeven bewijs dat er gen kleinste geheel getal bestaat:

Stel dat er toch een kleinste reëel getal zou bestaan. Noem dit r. Omdat r een reëel getal is, is $r-1$ dat ook. Maar $r-1 < r$, wat is tegenspraak is met de aanname dat r het kleinste reële getal zou zijn.

De aanname dat er toch een kleinste reëel getal bestaat leidt dus tot een tegenspraak. Deze aanname kan dus niet waar zijn, dus kan er geen kleinste reëel getal bestaan.

Blz. 12, it. 2: Elke som van even getallen is weer even. Dus de som van drie even getallen is even. Dus de som van drie even getallen kan nooit oneven zijn.

Blz. 12, it. 3: Stel toch. Dan is er een irrationaal getal, x, zodanig dat

$$x + \frac{m}{n} = \frac{k}{l}$$

met k, $l \neq 0$, m, $n \neq 0$ geheel. Maar dan

$$x = \frac{k}{l} - \frac{m}{n} = \frac{kn - ml}{ln}, \qquad nl \neq 0,$$

wat zou betekenen dat x een breuk is. Tegenspraak.

Blz. 12, it. 4: Stel toch. Schrijf de vergelijking als

$$(x+y)(x-y) = 1.$$

Om het product 1 te laten zijn, moeten beide factoren tegelijkertijd 1 of -1 zijn. Maar de stelsels $\{x+y=1, x-y=1\}$ en $\{x+y=-1, x-y=-1\}$ bezitten beide geen positieve geheeltallige oplossingen.

Blz. 12, it. 5: Stel toch. Dan

$$\left(\frac{p}{q}\right)^3 + \frac{p}{q} + 1 = 0.$$

voor gehele p, $q \neq 0$. Neem zonder beperking der algemeenheid ook maar even aan dat $\gcd(p,q) = 1$. Alles met q^3 vermenigvuldigen levert

$$p^3 + pq^2 + q^3 = 0$$

1. p en q beiden oneven. Dan linkerkant oneven. Dus rechterkant oneven. Tegenspraak.
2. p even en q oneven. Weer is de linkerkant oneven. Tegenspraak.
3. p oneven en q even. Weer is de linkerkant oneven. Tegenspraak.
4. p even en q even. Dit is in tegenspraak met de eerdere aanname dat $\gcd(p,q) = 1$.

Blz. 12, it. 6: Stel $p_1, p_2, \ldots, p_r$ zijn alle priemgetallen. Laat

$$P = (p_1 \times p_2 \times \cdots \times p_r) + 1$$

en laat p een priemdeler van P zijn. Dan kan p niet gelijk zijn aan één van de $p_1, p_2, \ldots, p_r$, want anders zou p een deler zijn van $P - p_1 p_2 \ldots p_r = 1$, wat onmogelijk is. [We maakten hier gebruik van het feit dat, als $p|A$ en $p|B$, dat dan ook $p|(A-B)$.] Dus p is een nieuw priemgetal dat verschilt van alle $p_1, p_2, \ldots, p_r$. Tegenspraak.

Het is een vaak voorkomend misverstand om te denken dat uit dit bewijs volgt dat P priem is maar P hoeft niet priem te zijn. Bijvoorbeeld, het product van de eerste 6 priemgetallen + 1 is gelijk aan

$$(2 \times 3 \times 5 \times 7 \times 11 \times 13) + 1 = 30031,$$

maar 30031 is geen priem, want $30031 = 59 \times 509$. Het bewijs gebruikt alleen het feit dat P deelbaar is door een priemgetal.

Blz. 12, it. 1.21: Hypothese:

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Te bewijzen:

$$1 + 2 + \cdots + n + (n+1) = \frac{(n+1)(n+2)}{2}.$$

Blz. 12, it. 1: Basis $n = 0$ klopt: $2^0 = 1 \leq 1 = 3^0$. Inductiehypothese: stel $2^n \leq 3^n$ is waar voor zekere $n \geq 0$. Beide kanten met 2 vermenigvuldigen levert $2^{n+1} \leq 2\cdot 3^n$ op. Nu geldt natuurlijk dat

$$2\cdot 3^n < 3\cdot 3^n = 3^{n+1}.$$

Beide ongelijkheden achter elkaar zetten levert $2^{n+1} < 3^{n+1}$ op.

Blz. 12, it. 2: Basis $n = 1$ klopt: $2\cdot 1 - 1 = 1^2$. Inductiehypothese: stel $1 + 3 + 5 + \cdots + (2n-1) = n^2$ is waar voor zekere $n \geq 0$. We moeten nu laten zien dat

$$1 + 3 + 5 + \cdots + (2n-1) + (2n+1) = (n+1)^2.$$

Als volgt: het gedeelte $1 + 3 + 5 + \cdots + (2n-1)$ in

$$1 + 3 + 5 + \cdots + (2n-1) + (2n+1)$$

kunnen we met behulp van de inductiehypothese schrijven als n^2, zodat we

$$n^2 + (2n+1)$$

krijgen. Het is gemakkelijk in te zien dat deze uitdrukking inderdaad gelijk is aan $(n+1)^2$. Klaar!

Blz. 12, it. 3: Geen uitwerking.

Blz. 12, it. 4: Geen uitwerking.

Blz. 12, it. 5: Geen uitwerking.

Blz. 12, it. 6: Basis $n = 4$ klopt: $4! = 24 \geq 16 = 2^4$. (Merk op dat $n = 4$ inderdaad het eerste natuurlijke getal is waarvoor de verlangde ongelijkheid op gaat.) Inductiehypothese: stel $n! \geq 2^n$ is waar voor zekere $n \geq 4$. Dan

$$\begin{aligned}(n+1)! &= (n+1)n! \\ &\geq (n+1)2^n && \text{[per inductie]} \\ &\geq 2 \cdot 2^n && \text{[immers, als } n \geq 4 \text{ dan } n+1 \geq 2\text{]} \\ &= 2^{n+1}.\end{aligned}$$

Blz. 12, it. 7: Basis $n = 0$ klopt: $0^3 + 2 \times 0 = 0$ en $3|0$. Stel de bewering is waar voor n. We gaan kijken of de bewering waar is voor $n+1$:

$$(n+1)^3 + 2(n+1) = n^3 + 3n^2 + 5n + 3$$
$$= \underbrace{n^3 + 2n}_{\substack{\text{deelbaar door} \\ 3 \text{ per inductie}}} + \underbrace{3n^3 + 3n + 3}_{\substack{\text{deelbaar} \\ \text{door } 3}}.$$

Als elke term van een som deelbaar is door drie, dan is de som zelf dat ook.

Blz. 12, it. 8: Basis $n = 0$ klopt: $(r+1)^0 = 1 = 0 \cdot r + 1$. Inductiehypothese: stel $(r+1)^n \geq nr + 1$, voor $r > -1$ is waar voor zekere $n \geq 0$. Dan

$$\begin{aligned}(r+1)^{n+1} &= (r+1)^n(r+1) \\ &\geq (nr+1)(r+1) && \text{[per inductie]} \\ &= nr^2 + nr + r + 1 \\ &= (n+1)r + 1 + nr^2 \\ &\geq (n+1)r + 1. && \text{[restterm } nr^2 \geq 0\text{]}\end{aligned}$$

De conditie $r > -1$ is een noodzakelijke voorwaarde: anders zou $r+1$ in $(r+1)^n$ kleiner of gelijk nul worden en dan moeten we in het bewijs rekening gaan houden met het feit dat negatieve getallen met oneven exponenten weer negatief worden. Dit feit blijkt bij nadere beschouwing inderdaad een bewijs van dit algemenere geval te obstrueren.

Blz. 13, it. 2a: $b_n = 2^n$.

Blz. 13, it. 3: Let $P(n)$ be the statement "Postage of n cents can be formed using 4-cent and 5-cent stamps".

REGULAR INDUCTION. Basis: $k = 12$. $P(12)$ is true, because 12 cents can be formed using three 4-cent stamps.

Induction: assume that $P(k)$ is true, that is, that postage of k cents can be formed using some combination of 4-cent and 5-cent stamps. Then to form postage of $k+1$ cents, there are two possibilities.

1. If any 4-cent stamps were used to get k cents of postage, take one of the 4-cent stamps and replace it with a 5-cent stamp. You now have $k+1$ cents of postage.
2. If only 5-cent stamps were used to get k cents of postage, since we know that $k \geq 12$, we have at LEAST three 5-cent stamps. (I.e. the smallest k that can only be formed using 5-cent stamps is 15.) So if you replace any three 5-cent stamps with four 4-cent stamps, you will have $k+1$ cents of postage.

STRONG INDUCTION. Basis: this proof requires four base cases.

$k = 12$**:** 12 cents of postage can be formed using three 4-cent stamps.

$k = 13$**:** 13 cents of postage can be formed using two 4-cent stamps and a 5-cent stamp.

$k = 14$**:** 14 cents of postage can be formed using a 4-cent stamp and two 5-cent stamps.

$k = 15$**:** 15 cents of postage can be formed using three 5-cent stamps.

Induction: for $k > 15$, assume that we can form postages of j cents, for every j between 12 and k, inclusive. Then, to form a postage for $k+1$ cents, use the same postage as for $k-3$ cents, but add a 4-cent stamp.

Blz. 13, it. 1.24: A: ja, -5: nee, (-6): ja, $(3+4)$: ja, $(3-4)$: nee, $3+4$: nee, $3+(F+4)$: nee, $(3+(G+4))$: ja, $(-(K+(3+A)))$: ja, $-(P+(P+4))$: nee

Blz. 14, it. 2: $l(\varphi) = 4o(\varphi) + 1$.

Blz. 14, it. 3: $l(\varphi) = 3o(\varphi) + 1$.

Blz. 14, it. 4: $l(\varphi) \leq 4o(\varphi) + 1$.

Blz. 14, it. 5: **Inductiebasis:** φ bevat nul operatoren. Als $o(\varphi) = 0$, dan is φ een atoom en dan per definitie $l(\varphi) = 1$. De ongelijkheid $l(\varphi) \leq 4o(\varphi) + 1$ klopt in dit geval. (Het is in feite een gelijkheid, maar dat maakt verder niet uit.)

Inductiestap: stel $l(\varphi) \leq 4o(\varphi) + 1$ is waar voor formules met minder dan $n > 1$ operatoren. Stel φ bezit n operatoren. Nu zijn er drie mogelijkheden: φ is van de vorm $(-\psi)$, φ is van de vorm $(\psi + \chi)$, of φ is van de vorm $(\psi \times \chi)$. Stel het laatste. Dan

$$\begin{aligned}l(\varphi) &= l[(\psi \times \chi)] \\ &= 3 + l(\psi) + l(\chi) && \text{[twee (,) en een } \times\text{]} \\ &\leq 3 + (4o(\psi) + 1) + (4o(\chi) + 1) && \text{[inductiehypothese]} \\ &= 4(o(\psi) + o(\chi) + 1) + 1 \\ &= 4o(\varphi) + 1 && [o(\varphi) = o(\psi) + o(\chi) + 1].\end{aligned}$$

Voor $\varphi = (-\psi)$, en $\varphi = (\psi + \chi)$ analoog.

Blz. 14, it. 6: Er *mogen* haakjes worden weglaten. Er kunnen dus expressies voorkomen waarbij geen enkel haakje is weggelaten. Het antwoord is dus hetzelfde als het antwoord in onderdeel 4.

Blz. 14, it. 7: $l(\varphi) \leq \max\{1, 4o(\varphi) - 1\}$.

Blz. 14, it. 1.26: Het inductie-argument houdt geen stand als $|A| = 2$. In dat geval bevatten B en C geen gemeenschappelijk element meer.

Blz. 14, it. 1.27: Deze opgave heeft geen eenvoudig antwoord. Google op "Interesting number paradox".

Blz. 15, it. 1: Non-constructief aspect: je kunt niet aangeven in welke doos meer dan twee voorwerpen zitten.

Blz. 15, it. 2: Zij n deelbaar. Dan is $n = km$ voor zekere $k, m < n$. Stel k en $m > \sqrt{n}$. Dan $n = km > \sqrt{n}\sqrt{n} = n$. Dat kan niet, dus één van k, m moet kleiner gelijk zijn aan $\sqrt{n}$. Non-constructief aspect: je kunt met dit bewijs niet aangeven welke deler van n kleiner of gelijk aan $\sqrt{n}$ is.

Blz. 15, it. 3: De grafiek van een veelterm is continu. Als de hoogste exponent oneven is, verdwijnt deze rechts naar oneindig en links naar min-oneindig. Dankzij de zogenaamde tussenwaardestelling (check Wikipedia) moet die grafiek ergens de lijn $y = 0$ snijden. Dat is een nulpunt. Non-constructief aspect: voor $n \geq 5$ valt niet precies uit te drukken waar dat nulpunt precies ligt. (Dat dat niet kan is onafhankelijk bewezen door Abel en Galois. Beide stierven overigens zeer jong,—26 en 20 resp.)

Blz. 15, it. 4: $n^3 - n = (n-1)n(n+1)$. Van drie opeenvolgende getallen is er precies één deelbaar door drie! Dus het product is deelbaar door drie.
Non-constructief aspect: je kunt niet aangeven welke van de drie $n-1$, n, $n+1$ deelbaar door drie is.

Blz. 15, it. 5: Neem één persoon, A. Ga na dat A (tenminste) drie personen moet kennen of, als dat niet het geval is, (tenminste) drie personen niet kent. (Dit is weer een toepassing van het pigeon hole principle.) Laten we aannemen dat A (tenminste) drie personen kent. Stel, dat zijn B, C, en D. Als geen van deze drie elkaar kent, zijn we klaar. In het ander geval zijn er twee van deze drie die elkaar kennen en ontstaat er samen met A een groepje dat elkaar kent, en zijn we ook klaar.

Als we aannemen dat A juist drie personen niet kent, dan verloopt de redenering hetzelfde, alleen dan met de woorden "kennen" en "niet kennen" omgedraaid.

Blz. 16, it. 1.29: A closer look at the slanted sides of the trapezoidal and triangular pieces shows that they cannot be aligned as implied in the above fallacious illustrations. In fact, they are the diagonals of two dissimilar rectangles of sizes 2×5 and 3×8, respectively, and hence have distinct slopes. But the difference of the ratios ($2/5 = 0.4$ versus $3/8 = 0.375$) is too small to be perceived by the eye.

Zoek verder op internet met "chessboard paradox," or "dissection Fallacy".

Blz. 22, it. 1: Ja: $\{a\}$ is het enige element.

Blz. 22, it. 2: Nee: de verzameling heeft 2 elementen, $\{a\}$ en $\{b\}$.

Blz. 22, it. 3: Ja: het enige element is de verzameling uit *2*.

Blz. 22, it. 4: Ja: $\{a\}$ is het enige element.

Blz. 22, it. 5: Ja, want de verzameling is gelijk aan $\{\{a, b\}\}$.

Blz. 23, it. 2.2:

1. waar
2. onzinnig (maar strikt genomen wel waar: a is geen verzameling, en heeft dus geen elementen; er is dus geen element van a dat *niet* in $\{a\}$ zit!)
3. idem
4. waar
5. onwaar
6. waar
7. onwaar
8. waar
9. waar
10. waar

Blz. 23, it. 1: Stel $\emptyset'$ speelt ook de rol van de lege verzameling. We moeten laten zien dat voor alle x: $x \in \emptyset \Leftrightarrow x \in \emptyset'$. Deze equivalentie is, vanwege de rol die $\emptyset$ en $\emptyset'$ spelen vervuld: zowel $\emptyset$ als $\emptyset'$ bevatten dezelfde elementen, namelijk: geen.

Blz. 23, it. 2: $\emptyset$ is niet echt bevat in $\emptyset$ omdat elk element van $\emptyset$ element is van $\emptyset$ en omgekeerd. $\emptyset$ is wel echt bevat in elke andere verzameling omdat elk element van $\emptyset$ element is van elke andere verzameling, maar niet omgekeerd.

Blz. 24, it. 2.4:

1. 'De lege verzameling is een deelverzameling van $\{a\}$.' Dit is waar.
2. 'De lege verzameling is een deelverzameling van $\{\emptyset\}$.' Dit is waar.
3. 'De lege verzameling is een element van $\{\emptyset\}$.' Dit is waar.
4. 'De lege verzameling is een element van de lege verzameling.' Dit is onwaar, want de lege verzameling bevat geen elementen.
5. 'De lege verzameling is geen element van de lege verzameling.' Dit is waar.
6. 'De lege verzameling is een element van $\{a\}$.' Dit is onwaar, want $\{a\}$ bevat alleen a als element.

Blz. 24, it. 2.5:

– *commutativiteit*: $x \in A \cup B$ desda $x \in A$ of $x \in B$, dat wil zeggen desda $x \in B$ of $x \in A$, dat wil zeggen desda $x \in B \cup A$. Dus $A \cup B = B \cup A$.

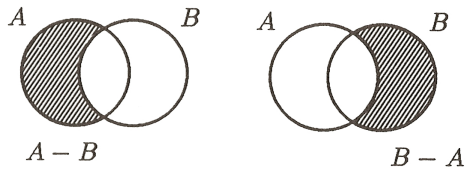
– *associativiteit*: $x \in A \cup (B \cup C)$ desda $x \in A$ of $x \in B \cup C$ desda $x \in A$ of $x \in B$ of $x \in C$, dat wil zeggen desda $x \in A \cup B$ of $x \in C$, dat wil zeggen desda $x \in (A \cup B) \cup C$. Dus $A \cup (B \cup C) = (A \cup B) \cup C$). □

Blz. 25, it. 2.7: 'A doorsneden met B" betekent niet hetzelfde als "B doorsneden met A". Het levert wel dezelfde uitkomst op, maar dat is iets anders. Bovendien willen we dat laatste nou juist bewijzen! Dat kan alleen maar door de definitie van doorsnijding twee keer in te vullen: één keer voor $A \cap B$ en één keer voor $B \cap A$, en dan in te zien dat we op hetzelfde uitkomen.

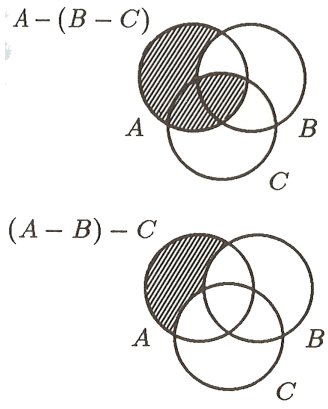
Blz. 25, it. 2.8:
- *idempotentie*: $x \in A \cap A$ desda $x \in A$ en $x \in A$, dus desda $x \in A$. Ergo $A \cap A = A$.
- *associativiteit*: $x \in A \cap (B \cap C)$ desda $x \in A$ en $x \in B \cap C$ desda $x \in A$ en $x \in B$ en $x \in C$, dat wil zeggen desda $x \in A \cap B$ en $x \in C$, dus desda $x \in (A \cap B) \cap C$. Dus $A \cap (B \cap C) = (A \cap B) \cap C$). □

Blz. 25, it. 2.9: De verschil-operatie is

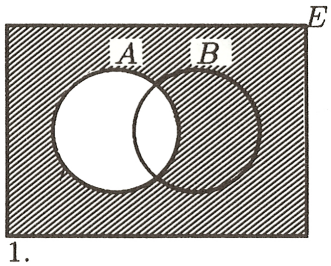
- *niet idempotent*, want $A - A = \emptyset$.
- *niet commutatief*, want $A - B$ is niet hetzelfde als $B - A$:



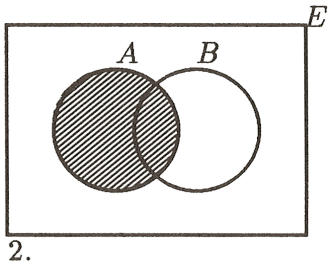
- *niet associatief*, want $A - (B - C)$ is niet hetzelfde als $(A - B) - C$:



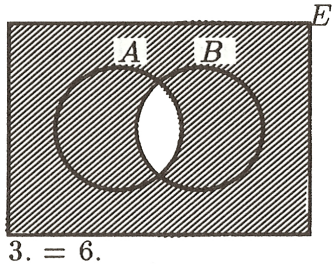
Blz. 25, it. 2.10: De juiste arceringen zijn als volgt:



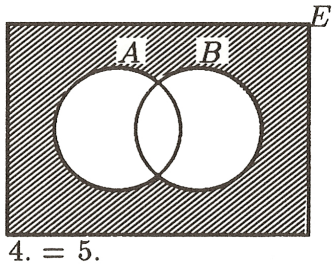
1.



2.



3. = 6.



4. = 5.

Hieruit blijkt dat de volgende wetten gelden: $A^c \cup B^c = (A \cap B)^c$ en $A^c \cap B^c = (A \cup B)^c$.

Blz. 26, it. 1: Onwaar: het linkerlid is $(A \cap B) \cup C$, het rechterlid is $(A \cap C) \cup B$. Het linker- en het rechterlid zijn nu zo geschreven dat je er makkelijk Venn-diagrammen van kunt tekenen. Als we dat doen, dan zien we dat het linker- en het rechterlid bijvoorbeeld in het gebied $B \setminus (A \cup C)$ ("alles van B wat niet in C ligt") verschillen. Kies dus als tegenvoorbeeld

$$A = C = \emptyset \text{ en } B = \{0\}.$$

(Iedere keuze met een gevulde $B \setminus (A \cup C)$ is goed, maar een minimaal tegenvoorbeeld verdient de voorkeur.) Dan

$$(A \cap B) \cup C = (\emptyset \cap \{0\}) \cup \emptyset = \emptyset,$$
$$(A \cup B) \cap (B \cup C) = (\emptyset \cup \{0\}) \cap (\{0\} \cup \emptyset) = \{0\} \cap \{0\} = \{0\}.$$

En $\{0\} \neq \emptyset$.

Blz. 26, it. 2: Waar. Bewijs: we bewijzen

$$(A \cup B) \cap C \subseteq (A \cap C) \cup (B \cap C) \text{ en}$$
$$(A \cup B) \cap C \supseteq (A \cap C) \cup (B \cap C).$$

(⊆): Zij $x \in (A \cup B) \cap C$ willekeurig. Dan per definitie van "$\cap$" $x \in A \cup B$ en $x \in C$. Per definitie van "$\cup$" $x \in A$ of $x \in B$. Daarnaast nog steeds $x \in C$. Maar dat is hetzelfde als te zeggen:

$$x \in A \text{ en } x \in C, \text{ of } x \in B \text{ en } x \in C.$$

Deze Nederlandse zin op dezelfde manier in enkele stappen weer terug herschrijven in

verzamelingstheoretische symbolen geeft $x \in (A \cap C) \cup (B \cap C)$. Omdat x willekeurig gekozen was, geldt

$$(A \cup B) \cap C \subseteq (A \cap C) \cup (B \cap C). \qquad (9)$$

($\supseteq$): Zij $x \in (A \cap C) \cup (B \cap C)$ willekeurig. Dan per definitie van "$\cup$" $x \in A \cap C$ of $x \in B \cap C$. Twee keer de definitie van "$\cap$" toepassen geeft

$x \in A$ en $x \in C$, of $x \in B$ en $x \in C$.

Maar dat is hetzelfde als zeggen:

$x \in A$ of $x \in B$, en $x \in C$.

Deze Nederlandse zin op dezelfde manier in enkele stappen weer terug herschrijven in verzamelingstheoretische symbolen geeft $x \in (A \cup B) \cap C$. Omdat x willekeurig gekozen was, geldt

$$(A \cap C) \cup (B \cap C) \subseteq (A \cup B) \cap C.$$

Samen met (9) geeft dit de gevraagde gelijkheid.

Dit bewijs kan sneller door ($\subseteq$) en ($\supseteq$) tegelijkertijd te bewijzen met $\Leftrightarrow$, of door gebruik te maken van reeds bewezen verzamelingstheoretische identiteiten, zoals de distributieve wetten voor "$\cap$" en "$\cup$". Het punt is dat je een rechtstreeks bewijs moet kunnen geven. (En dit was een rechtstreeks bewijs.)

Blz. 26, it. 3: Onwaar. Als je Venn-diagrammen tekent, dan zie je dat het linker- en rechterlid in het gebied $B \cap C$ verschillen. In een tegenvoorbeeld moet $B \cap C$ dus gevuld zijn, de rest maakt niet uit, en kiezen we dus, om het tegenvoorbeeld zo klein mogelijk te houden, leeg:

$$A = \emptyset, B = C = \{0\}.$$

$$(A \cup B) \setminus C = (\emptyset \cup \{0\}) \setminus \{0\} = \{0\} \setminus \{0\} = \emptyset,$$
$$(A \setminus C) \cup B = (\emptyset \setminus \{0\}) \cup \{0\} = \emptyset \cup \{0\} = \{0\}.$$

En $\{0\} \neq \emptyset$.

Blz. 26, it. 4: Waar. In een Venn-diagram is dit een klein gebiedje: $A \cap B \cap C^c$. We geven een bewijs dat wat "sneller" verloopt als het bewijs van de gelijkheid in (bijvoorbeeld) onderdeel 2.

Zij $x \in U$ willekeurig. (U is het universum.)

$$\begin{aligned}
& x \in (A \cap B) \setminus C \\
\Leftrightarrow\ & x \in A \cap B \text{ en } x \notin C \\
\Leftrightarrow\ & x \in A \text{ en } x \in B \text{ en } x \notin C \\
\Leftrightarrow\ & x \in A \text{ en } x \notin C \text{ en } x \in B \\
\Leftrightarrow\ & x \in A \setminus C \text{ en } x \in B \\
\Leftrightarrow\ & (A \setminus C) \cap B.
\end{aligned}$$

Dit is een prima bewijs, zolang alle bi-implicaties $\Leftrightarrow$ maar terecht opgeschreven zijn, i.e., zolang elk tweetal opeenvolgende beweringen ook echt logisch gelijkwaardig is. Soms kun je hierin, om het wat dramatisch uit te drukken, "vreselijke uitglijders" maken, dat wil zeggen dat je een "$\Leftrightarrow$" opschrijft waar in werkelijkheid slechts "$\Rightarrow$" of "$\Leftarrow$" geldt.

Blz. 26, it. 5: Waar. Zij $x \in U$ willekeurig.

$$\begin{aligned}
& x \in A \setminus (B \cup C) \\
\Leftrightarrow\ & x \in A \text{ en } x \notin B \cup C \\
\Leftrightarrow\ & x \in A \text{ en niet } x \in B \cup C \\
\Leftrightarrow\ & x \in A \text{ en niet } (x \in B \text{ of } x \in C) \\
\Leftrightarrow\ & x \in A \text{ en niet } x \in B \text{ en niet } x \in C \\
\Leftrightarrow\ & x \in A \text{ en } x \notin B \text{ en } x \notin C \\
\Leftrightarrow\ & x \in A \text{ en } x \notin B, \text{ en } x \in A \text{ en } x \notin C \\
\Leftrightarrow\ & x \in A \setminus B, \text{ en } x \in A \setminus C \\
\Leftrightarrow\ & x \in (A \setminus B) \cap (A \setminus C).
\end{aligned}$$

Blz. 26, it. 6: Waar. Zij $x \in U$ willekeurig.

$$\begin{aligned}
& x \in A \setminus (B \cap C) \\
\Leftrightarrow\ & x \in A \text{ en } x \notin B \cap C \\
\Leftrightarrow\ & x \in A \text{ en niet } x \in B \cap C \\
\Leftrightarrow\ & x \in A \text{ en niet } (x \in B \text{ en } x \in C) \\
\Leftrightarrow\ & x \in A \text{ en (niet } x \in B \text{ of niet } x \in C) \\
\Leftrightarrow\ & x \in A \text{ en } (x \notin B \text{ of } x \notin C) \\
\Leftrightarrow\ & x \in A \text{ en } x \notin B, \text{ of } x \in A \text{ en } x \notin C \\
\Leftrightarrow\ & x \in A \setminus B \text{ of } x \in A \setminus C \\
\Leftrightarrow\ & x \in (A \setminus B) \cup (A \setminus C).
\end{aligned}$$

Blz. 26, it. 1: Stel, er zijn vier verzamelingen: A, B, C, em D. Verzameling A wordt gerepresenteerd door de cirkel linksboven; verzameling B wordt gerepresenteerd door de cirkel rechtsboven; verzameling C wordt gerepresenteerd door de cirkel linksonder, etc.

Merk nu op dat de doorsnijding van elk tweetal tegenoverliggende verzamelingen bevat is in de vereniging van de twee andere tegenoverliggende verzamelingen:

$$A \cap D \subseteq B \cup C \quad \text{en} \quad B \cap C \subseteq A \cup D.$$

Dit hoeft in het algemeen niet zo te zijn: vier verzamelingen kunnen zich best zo verhouden dat bijvoorbeeld A en D gemeenschappelijke elementen bevat die buiten B en C liggen.

Blz. 26, it. 2: Voorbeelden te vinden op internet. (Zoek met voor de hand liggende termen.)

Blz. 26, it. 3: Ja, enig internetten levert op dat zoiets kan, onder andere met een constructie van John Venn zelf, door elke volgende afbeelding van een verzameling langs de rand te leggen van een vorige verzameling.

Alternatieve constructies werden voorgesteld door A. W. F. Edwards (tandwiel constructie) en Branko Grünbaum (overlappende sinusfiguren met steeds toenemende frequentie).

Blz. 27, it. 1: Stel $A \subseteq B$. TBW: $A \cup B = B$. Eerst $A \cup B \subseteq B$. Stel $x \in A \cup B$ willekeurig. Dan $x \in A$ of $x \in B$.

Als $x \in B$ zijn we klaar. Als $x \in A$ gebruiken we het feit $A \subseteq B$ om te concluderen dat $x \in B$.

Stel omgekeerd dat $A \cup B = B$. TBW: $A \subseteq B$. Stel hiertoe $x \in A$ willekeurig. Dan zeker $x \in A$ of $x \in B$. Dus $x \in A \cup B$. Omdat $A \cup B = B$ geldt meteen $x \in B$.

Blz. 27, it. 2: Stel $A \subseteq B$. TWB: $A \cap B = A$. Eerst $A \subseteq A \cap B$. Stel $x \in A$. Omdat $A \subseteq B$ ook $x \in B$. Dus $x \in A$ en $x \in B$, dus $x \in A \cap B$. Omdat x willekeurig was, geldt nu $A \subseteq A \cap B$. De omgekeerde inclusie $A \cap B \subseteq A$ is triviaal.

Stel omgekeerd $A \cap B = A$. TBW: $A \subseteq B$. Stel hiertoe $x \in A$ willekeurig. Vanwege $A \cap B = A$ ook $x \in B$. Omdat x willekeurig was, geldt $A \subseteq B$.

Blz. 27, it. 3: Stel $A \subseteq B^c$. TBW: $A \cap B = \emptyset$. Stel toch $A \cap B \neq \emptyset$. Dan is er een x, zo dat $x \in A \cap B$. Dan $x \in A$ en $x \in B$. Vanwege $A \subseteq B^c$ geldt nu ook $x \in B^c$ dus $x \notin B$. Tegenspraak. (Want eerder concludeerden we $x \in B$.) Dus zo'n $x \in A \cap B$ kan niet bestaan. Dus $A \cap B = \emptyset$.

Stel omgekeerd dat $A \cap B = \emptyset$. TBW: $A \subseteq B^c$. Stel $x \in A$ willekeurig. Omdat $A \cap B = \emptyset$ kan het niet zo zijn dat $x \in B$. Dus $x \notin B$. Dus $x \in B^c$.

Blz. 27, it. 4: Stel eerst $A^c \subseteq B$. TBW: $A \cup B = X$. De inclusie $A \cup B \subseteq X$ hoeft uiteraard niet te worden bewezen, want X is het universum. Laten we $X \subseteq A \cup B$ bewijzen. Laat daartoe $x \in X$ willekeurig. Er zijn twee mogelijkheden: $x \in A$ of $x \notin A$. Als $x \in A$ dan ook $x \in A \cup B$ en zijn we klaar. Als $x \notin A$ dan $x \in A^c$. Omdat $A^c \subseteq B$ was aangenomen dus ook $x \in B$. Dus $x \in A \cup B$. In beide gevallen ($x \in A$ of $x \notin A$) dus $x \in A \cup B$.

Stel omgekeerd dat $A \cup B = X$. TBW: $A^c \subseteq B$. Neem daartoe $x \in A^c$ willekeurig. We proberen x in B te "praten". Omdat $x \in X = A \cup B$ moet $x \in A$ of $x \in B$. Maar omdat $x \in A^c$ geldt $x \notin A$. Dus moet $x \in B$. omdat x willekeurig gekozen is, is het gevraagde bewezen.

Blz. 27, it. 5: Stel $A \subseteq B$. TBW: $B^c \subseteq A^c$. Neem $x \in B^c$ willekeurig. Dus $x \notin B$. Vanwege onze aanname geldt "voor alle y: $y \in A \Rightarrow y \in B$". De contrapositie daarvan is "voor alle y: $y \notin B \Rightarrow y \notin A$.". In het bijzonder geldt de contrapositie voor $y = x$. De contrapositie geeft dus $x \notin A$, dus $x \in A^c$.

Stel omgekeerd dat $B^c \subseteq A^c$. TBW: $A \subseteq B$. Neem $x \in A$ willekeurig. Vanwege onze aanname geldt

$$\begin{aligned} & \text{voor alle } y : y \in B^c \Rightarrow y \in A^c \\ \Leftrightarrow\ & \text{voor alle } y : y \notin B \Rightarrow y \notin A \\ \Leftrightarrow\ & \text{voor alle } y : \text{niet } y \in B \Rightarrow \text{niet } y \in A \\ \Leftrightarrow\ & \text{voor alle } y : y \in A \Rightarrow y \in B \qquad \text{(contrapositie)} \end{aligned}$$

In het bijzonder geldt de contrapositie voor $y = x$. Dus $x \in B$. Omdat x willekeurig was volgt het gestelde.

Blz. 27, it. 6: Analoog aan het antwoord van de vorige opgave.

Blz. 27, it. 1: $\bigcup\{A, B\}$.

Blz. 27, it. 2:

$$\begin{aligned} & x \in \bigcup\{(0,z) \mid z \in \mathbb{R}\} \\ \Leftrightarrow\ & \text{er is een } z \in \mathbb{R}, \text{ zó dat } x \in (0,z) \\ \Leftrightarrow\ & x > 0. \end{aligned}$$

Dus $\bigcup\{(0,z) \mid z \in \mathbb{R}\} = (0, \rightarrow)$.

$$\begin{aligned} & x \in \bigcap\{(0,z) \mid z \in \mathbb{R}\} \\ \Leftrightarrow\ & \text{voor alle } z \in \mathbb{R} \text{ met } z > 0 \text{ geldt dat } x \in (0,z). \end{aligned}$$

Aan deze eis kan redelijkerwijs geen enkele $x \in \mathbb{R}$ voldoen. Dus $\bigcap\{(0,z) \mid z \in \mathbb{R}\} = \emptyset$.

Blz. 27, it. 1: Stel $x \in A_j$ willekeurig. Dan is er een $i \in I$ zodanig dat $x \in A_i$, namelijk $i = j$. Per definitie van $\bigcup_{i \in I} A_i$ geldt nu $x \in \bigcup_{i \in I} A_i$.

Stel $x \in \bigcap_{i \in I} A_i$ willekeurig. Per definitie geldt nu dat $x \in A_i$ voor alle $i \in I$. Dus ook voor $i = j$. Dus $x \in A_j$.

Blz. 28, it. 2:

$$\begin{aligned} & x \in \left(\bigcup_{i \in I} A_i\right) \cup \left(\bigcup_{i \in I} B_i\right) \\ \Leftrightarrow\ & x \in \bigcup_{i \in I} A_i \text{ of } x \in \bigcup_{i \in I} B_i \\ \Leftrightarrow\ & \text{er is een } i \in I \text{ z.d.d. } x \in A_i \\ & \qquad \text{of er is een } i \in I \text{ z.d.d. } x \in B_i \\ (!)\ \Leftrightarrow\ & \text{er is een } i \in I \text{ z.d.d. } (x \in A_i \text{ of } x \in \cup B_i) \\ \Leftrightarrow\ & \text{er is een } i \in I \text{ z.d.d. } x \in A_i \cup B_i \\ \Leftrightarrow\ & x \in \bigcup_{i \in I} (A_i \cup B_i). \end{aligned}$$

De overgang bij het uitroepen is niet helemaal triviaal. Je moet daar namelijk inzien dat één lopende index eigenlijk genoeg is om de statement te kunnen maken.

De andere gelijkheid kan analoog worden bewezen.

Blz. 28, it. 5: Neem x willekeurig.

$$\begin{aligned} & x \in \bigcup_{i \in I} (X \setminus A_i) \\ \Leftrightarrow\ & \text{er is een } i \in I \text{ z.d.d. } x \in X \setminus A_i \\ \Leftrightarrow\ & \text{er is een } i \in I \text{ z.d.d. } x \in X \text{ en } x \notin A_i \\ (!)\ \Leftrightarrow\ & x \in X \text{ en er is een } i \in I \text{ z.d.d. } x \notin A_i \\ \Leftrightarrow\ & x \in X \text{ en niet voor alle } i \in I \text{ geldt } x \in A_i \\ \Leftrightarrow\ & x \in X \text{ en niet } x \in \bigcap_{i \in I} A_i \\ \Leftrightarrow\ & x \in X \text{ en } x \notin \bigcap_{i \in I} A_i \\ \Leftrightarrow\ & x \in X \setminus \bigcap_{i \in I} A_i. \end{aligned}$$

De overgang bij het uitroepen is niet helemaal triviaal. Je moet daar namelijk inzien dat de bewering $x \in X$ niet afhankelijk is van de index i die over I loopt, dus buiten de "er is een $i \in I$"-kwalificatie kan worden gehaald.

Blz. 28, it. 5: Analoog aan het bewijs van Item 5.

Blz. 28, it. 2.22: 1. $\emptyset$

2. $\emptyset$

3. $\{\emptyset\}$

4. $\{\emptyset, \{\emptyset\}\}$

5. $\{\emptyset\}$

6. $\{\emptyset, \{\emptyset\}\}$

Blz. 28, it. 2.24: Laat $A \subseteq B$. Neem een willekeurige $V \in 2^A$, dan is $V \subseteq A$. Vanwege het gegeven is dus ook $V \subseteq B$ (teken zo nodig een plaatje) met andere woorden $V \in 2^B$. Derhalve is $2^A \subseteq 2^B$. □

Blz. 28, it. 2.25: Stel $A \subseteq B$. Te bewijzen dat elke functie van A naar $\{0,1\}$ ook een functie van B naar $\{0,1\}$ is. Laat

$$s : A \to \{0,1\}$$

zo'n functie zijn. (De letter "s" representeert "subset".) Deze functie duidt een bepaalde deelverzameling $A' \subseteq A$ aan, en beeldt elementen $a \in A$ af op 1 als en slechts als $a \in A'$. De functie s is nu uit te breiden naar een functie

$$s' : B \to \{0,1\}$$

door

$$s'(b) = \begin{cases} s(b) & \text{als } b \in A \\ 0 & \text{anders.} \end{cases}$$

Hiermee hebben we aangetoond dat, als $A \subseteq B$, elke functie van A naar $\{0,1\}$ conservatief kan worden uitgebreid naar een functie van B naar $\{0,1\}$.

Hebben we nu aangetoond dat elke functie van A naar $\{0,1\}$ ook daadwerkelijk een functie van B naar $\{0,1\}$ ***is***? Dit ligt wat subtieler, maar het antwoord is bevestigend. Een functie is namelijk een voorschrift dat vastlegt waar elementen uit een domein, in dit geval B, op worden afgebeeld. De functie s legt dat voor elementen buiten A (maar binnen B) impliciet vast door deze op nul af te beelden.

Blz. 28, it. 2.26: Als $2^A \subseteq 2^B$, dan is elk element van 2^A element van 2^B. Aangezien $A \subseteq A$ en daarmee $A \in 2^A$, volgt uit het gegeven dat $A \in 2^B$. Dus $A \subseteq B$. □

Blz. 28, it. 2.27:

1. $2^{2^\emptyset} = 2^{\{\emptyset\}} = \{\{\emptyset\}, \emptyset\}$

2. $2^{2^{\{a\}}} = 2^{\{\{a\},\emptyset\}} = \{\{\{a\},\emptyset\}, \{\{a\}\}, \{\emptyset\}, \emptyset\}$

3. Het wordt nu een beetje zinloos om een antwoord uit te schrijven. We moeten gaan nadenken over de *structuur* van het antwoord. Allereerst kunnen we het aantal elementen gaan bepalen: één keer een machtsverzameling nemen, geeft een verzameling ter grootte $2^1 = 2$. Nog een keer een machtsverzameling nemen, geeft een verzameling ter grootte $2^2 = 4$. Nog een keer geeft $2^4 = 16$.

$$\begin{aligned} 2^{2^{2^{\{a\}}}} &= 2^{\{\{\{a\},\emptyset\},\{\{a\}\},\{\emptyset\},\emptyset\}} \\ &= \{\{\{\{a\},\emptyset\},\{\{a\}\},\{\emptyset\},\emptyset\}, \\ &\quad \{\{\{a\},\emptyset\},\{\{a\}\},\{\emptyset\}\},\{\{\{a\},\emptyset\},\{\{a\}\},\emptyset\}, \\ &\quad \{\{\{a\},\emptyset\},\{\emptyset\},\emptyset\},\{\{\{a\}\},\{\emptyset\},\emptyset\}, \\ &\quad \{\{\{a\},\emptyset\},\{\{a\}\}\},\{\{\{a\},\emptyset\},\{\emptyset\}\}, \\ &\quad \{\{\{a\},\emptyset\},\emptyset\}, \\ &\quad \{\{\{a\}\},\{\emptyset\}\},\{\{\{a\}\},\emptyset\},\{\{\emptyset\},\emptyset\}, \\ &\quad \{\{\{a\},\emptyset\}\},\{\{\{a\}\}\},\{\{\emptyset\}\},\{\emptyset\},\emptyset\} \end{aligned}$$

   Laten we eens doorgaan. nog een keer een machtsverzameling nemen, zou een verzameling ter grootte $2^{16} = 65536$ geven. Dan nóg een keer een machtsverzameling nemen, zou een verzameling ter grootte $2^{65536} \approx 2 \cdot 10^{19727}$ geven. Ter vergelijking: de grootte van het observeerbare heelal wordt geschat tussen de $4 \cdot 10^{79}$ en 10^{81} atomen.

4.

$$\begin{aligned} 2^{2^{\{a,b\}}} &= 2^{\{\{a,b\},\{a\},\{b\},\emptyset\}} \\ &= \{\{\{a,b\},\{a\},\{b\},\emptyset\}, \\ &\quad \{\{a,b\},\{a\},\{b\}\},\{\{a,b\},\{a\},\emptyset\}, \\ &\quad \{\{a,b\},\{b\},\emptyset\},\{\{a\},\{b\},\emptyset\}, \\ &\quad \{\{a,b\},\{a\}\},\{\{a,b\},\{b\}\},\{\{a,b\},\emptyset\}, \\ &\quad \{\{a\},\{b\}\},\{\{a\},\emptyset\},\{\{b\},\emptyset\}, \\ &\quad \{\{a,b\}\},\{\{a\}\},\{\{b\}\},\{\emptyset\},\emptyset\} \end{aligned}$$

Blz. 29, it. 2.28: $\{1,2,3\}^3$ heeft $3 \cdot 3 \cdot 3 = 3^3 = 27$ elementen. Immers elk element van deze verzameling is een tripel waarbij voor elk lid van zo'n tripel 3 onafhankelijke keuzen zijn.

Blz. 31, it. 3.1: $\text{dom}(R) = \{b,d,e\}$ $\quad$ $\text{rng}(R) = \{c,d,e\}$

Blz. 32, it. 3.2: We kunnen $2 > 1$ opvatten als $\langle 2,1\rangle \in R$, zodat $R \subset \mathbb{N} \times \mathbb{N}$ de relatie is op $\mathbb{N}$ die formeel met $>$ overeenkomt.

Blz. 32, it. 3.3: Bijvoorbeeld:

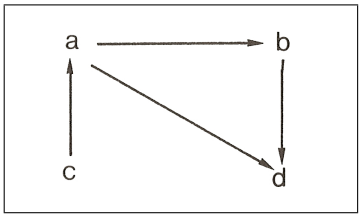
Deze relatie is niet reflexief want b, bijvoorbeeld, verwijst niet naar zichzelf. De relatie is ook niet irreflexief, want er is een element, namelijk a, dat naar zichzelf verwijst.

Blz. 32, it. 3.4: Laat de relatie $R \subseteq A \times A$ asymmetrisch zijn. Dan moet R wel irreflexief zijn, want anders zou er een $a \in A$ bestaan waarvoor $\langle a,a\rangle \in R$. Maar dat kan niet vanwege de asymmetrie van R: dan zou immers niet alleen het paar $\langle a,a\rangle$ in R zitten, maar ook het omgekeerde paar (dat is namelijk weer $\langle a,a\rangle$).

Blz. 32, it. 3.5: Bijvoorbeeld:

De relatie is *niet symmetrisch*, want $\langle a,b\rangle \in R$, maar $\langle b,a\rangle \notin R$. De relatie is ook niet *asymmetrisch*, want $\langle c,d\rangle \in R$ en ook $\langle d,c\rangle \in R$.

Blz. 33, it. 3.6: Bijvoorbeeld:



De relatie is niet *transitief*, want $\langle c,a\rangle \in R$ en $\langle a,d\rangle \in R$, maar $\langle c,d\rangle \notin R$. De relatie is daarnaast ook niet *in*transitief, want $\langle a,b\rangle \in R$, $\langle b,d\rangle \in R$ en ook $\langle a,d\rangle \in R$.

Blz. 33, it. 3.7: 1. Stel R is asymmetrisch. R is alleen dan *niet* anti-symmetrisch wanneer er een paar $\langle a,b\rangle$ bestaat zodat $\langle a,b\rangle \in R$, $\langle b,a\rangle \in R$ en $a \neq b$. De combinatie van de eerste twee eisen is echter uitgesloten door de asymmetrie van R. R moet dus wel anti-symmetrisch zijn. □

2. De relatie $\leq$ op $\mathbb{N}$ is anti-symmetrisch, maar niet asymmetrisch.

Blz. 33, it. 3.8: We kiezen in deze opdracht de elementen B, C en D zonder verdere vermelding steeds in 2^A. De relatie $\subseteq$ is

– *reflexief*: voor elke B geldt $B \subseteq B$ per definitie.

– *anti-symmetrisch*: De eigenschap houdt in dat voor elke verzameling B en C geldt: als $B \subseteq C$ en $C \subseteq B$, dan $B = C$. Dit volgt direct uit het extensionaliteitsaxioma en de definitie van $\subseteq$.

– *transitief*: We moeten bewijzen dat als $B \subseteq C$ en $C \subseteq D$, dan ook $B \subseteq D$. Laat dus $B \subseteq C$ en $C \subseteq D$. Dan geldt dat elk element van B ook in C zit, en elk element van C in D zit. Kortom elk element van B is ook element van D, ergo $B \subseteq D$. □

De relatie $\subset$ is

– *irreflexief*: Geen enkele verzameling is een *echte* deelverzameling van zichzelf.

– *asymmetrisch*: Als $B \subset C$, dan is niet elk element van C lid van B, en dus is $C \notin B$.

– *transitief*: Als $B \subset C$ en $C \subset D$, dan ook $B \subset D$. De lezer gelieve dit zelf na te gaan. □

Blz. 33, it. 3.9: De relatie $\leq$ op $\mathbb{N}$ is

– *reflexief*: Want voor elk natuurlijk getal n geldt $n \leq n$.

– *transitief*: Als $k \leq m$ en $m \leq n$ dan $k \leq n$

– *anti-symmetrisch*: Als $m \leq n$ en $n \leq m$ dan $m = n$.

Blz. 33, it. 3.10: Laat R een strikte partiële orde op A zijn. Beschouw nu de verzameling $R' = R \cup \{\langle a,a\rangle \mid a \in A\}$. Dan is $R \subseteq R'$, en R' is een partiële orde want is:

– *reflexief.* Dit volgt direct uit de definitie van R'.

– *anti-symmetrisch.* Uit de asymmetrie van R volgt dat als $\langle a,b\rangle$ en $\langle b,a\rangle$ in R' zitten, dan $\langle a,b\rangle \notin R$. Dus $a = b$.

– *transitief.* Dit volgt uit de transitiviteit van R.

Blz. 33, it. 3.11: De letters f en n. Bijvoorbeeld: $g : \mathbb{Z} \to \mathbb{Z} : x \mapsto x^2$, of $h : \mathbb{Z} \to \mathbb{Z} : y \mapsto y^2$. De letter $\mathbb{Z}$ kan niet worden veranderd.

Blz. 34, it. 1: Een geordend paar is één entiteit.

Blz. 34, it. 2: De functie f is een deelverzameling van

$$\mathbb{N} \times (\mathbb{N} \times \mathbb{N})$$

als f beschouwd wordt als relatie.

Blz. 35, it. 1: Binnen elk vierkant rond de oorsprong ter grootte $2n \times 2n$ bevinden zich eindig veel start-configuraties. Zet de nu inhoud van die vierkanten achter elkaar en je hebt een aftelling van start-configuraties. (Om dubbeltelling te voorkomen sommeer je voor vierkant $2n$ alleen configuraties die punten hebben buiten vierkant $2(n-1)$. Het voorkomen van dubbeltellingen is overigens niet nodig bij aftellingen.)

Blz. 35, it. 2: Als een populatie uitsterft, stabiliseert, of zich gaat herhalen blijft deze begrensd, en groeit dus niet. Omgekeerd, als een populatie niet groeit, zal deze begrensd blijven. Op een begrensd vlak zijn slechts eindig veel populaties mogelijk. Op enig moment zal er dus herhaling optreden.

Blz. 36, it. 3.14: 1. Geen functie, want een persoon kan gekoppeld zijn aan twee ouders.

2. Wel een functie, want een persoon is steeds gekoppeld aan precies één verzameling ouders (die afhankelijk van de omstandigheden eventueel atomair of leeg kan zijn).

Blz. 36, it. 3.15: Ja, de lege verzameling zelf!

Blz. 36, it. 3.16:

$$\begin{aligned}\{c,d\}^{\{a,b\}} = \{&\{\langle a,c\rangle,\langle b,c\rangle\},\{\langle a,c\rangle,\langle b,d\rangle\},\\ &\{\langle a,d\rangle,\langle b,c\rangle\},\{\langle a,d\rangle,\langle b,d\rangle\}\}\end{aligned}$$

Blz. 36, it. 1:

$$\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\} \in X$$

als

$$\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\} \subseteq X.$$

Dus moeten we controleren dat

$$\emptyset \in X, \quad \{\emptyset\} \in X, \qquad \{\emptyset, \{\emptyset\}\} \in X.$$

De waarheid van de drie laatste beweringen kan weer recursief gecontroleerd worden.

Blz. 36, it. 2:

$$f[A] = f[\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}] = \{0, 1, 2\}.$$

Blz. 36, it. 3:

$$f[=(A) = f(\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}) = 3.$$

Blz. 37, it. 3.18: 1. $f(a) = b$

2. $f(f(a)) = c$

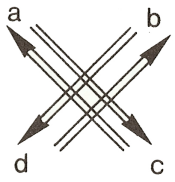
3. $f(f(f(a))) = d$

4. $f(f(f(f(a)))) = a$

5. $f(b) = c$

6. $f(f(f(f(b)))) = b$

Blz. 37, it. 3.19: Het plaatje wordt nu als volgt:



Blz. 37, it. 1: Niet waar: neem domein $X = \{0, 1\}$, co-domein $Y = \{2\}$, $f = \{(0, 2), (1, 2)\}$, $A' = \{0\}$, $A'' = \{1\}$. Dan

$$f[A' \cap A''] = f(\{0\} \cap \{1\}) = f[\emptyset] = \emptyset,$$
$$f[A'] \cap f[A''] = f[\{0\}] \cap f[\{1\}] = \{2\} \cap \{2\} = \{2\}.$$

Aangezien $\emptyset \neq \{2\}$ geldt in het algemeen niet $f[A' \cap A''] = f[A'] \cap f[A'']$. (Een plaatje tekenen helpt zeer.) Ons tegenvoorbeeld is nu af.

Wat wel geldt is

$$f[A' \cap A''] \subseteq f[A'] \cap f[A''].$$

Kijk maar:

$$\begin{aligned}
& y \in f[A' \cap A''] \\
\Leftrightarrow\ & \text{er is een } x \in A' \cap A'' \text{ zo dat } f(x) = y \\
(!)\ \Rightarrow\ & \text{er is een } x' \in A'' \text{ zo dat } f(x') = y \text{ en er is een } x'' \in A'' \text{ zo dat } f(x'') = y \\
& \text{(namelijk: neem } x' =_{Def} x \text{ en } x'' =_{Def} x) \\
\Leftrightarrow\ & y \in f[A'] \text{ en } y \in f[A''] \\
\Leftrightarrow\ & y \in f[A'] \cap f[A''].
\end{aligned}$$

Zie je waarom je implicatie "$\Leftarrow$" geen bi-implicatie "$\Leftrightarrow$" is? Omdat in de omgekeerde richting je maar moet afwachten of er één $x \in A' \cap A''$ de beide rollen van de niet noodzakelijk dezelfde x' en x'' kan spelen. In het door ons geconstrueerde tegenvoorbeeld hebben we er natuurlijk voor gezorgd dat dat juist mislukt.

Blz. 37, it. 2: Waar:

$$\begin{aligned}
& y \in f[A' \cup A''] \\
\Leftrightarrow\ & \text{er is een } x \in A' \cup A'' \text{ zo dat } f(x) = y \\
\Leftrightarrow\ & \text{er is een } x' \in A'' \text{ zo dat } f(x') = y \text{ of er is een } x'' \in A'' \text{ zo dat } f(x'') \\
\Leftrightarrow\ & y \in f[A'] \text{ of } y \in f[A''] \\
\Leftrightarrow\ & y \in f[A'] \cup f[A''].
\end{aligned}$$

Merk op dat hier wel overal bi-implicaties mogen staan. De tweede bi-implicatie is het meest kritisch. De implicatie "$\Leftarrow$" daarvan is geldig omdat voor x' en x'' gewoon weer x kan worden genomen. De implicatie "$\Rightarrow$" is geldig omdat voor x het element x' (als dat zo uitkomt) of x'' (als dat zo uitkomt) kan worden gekozen. Zolang die x maar uit $A' \cup A''$ komt. Maar dat zit voor wat betreft x' en x'' dus wel goed.

Blz. 37, it. 3: Geen uitwerking.

Blz. 37, it. 4: Werken met inverse beelden $f^{-1}[\cdot]$ is altijd makkelijker dan werken met beelden $f(\cdot)$. Deze identiteit is trouwens waar, wat meestal het geval is met inverse beelden. Zij $x \in X$ willekeurig.

$$\begin{aligned}
& x \in f^{-1}[B' \cap B''] \\
\Leftrightarrow\ & f(x) \in B' \cap B'' \\
\Leftrightarrow\ & f(x) \in B' \text{ en } f(x) \in B'' \\
\Leftrightarrow\ & x \in f^{-1}[B'] \text{ en } x \in f^{-1}[B''] \\
\Leftrightarrow\ & x \in f^{-1}[B'] \cap f^{-1}[B''].
\end{aligned}$$

Blz. 38, it. 5: Waar. Het bewijs verloopt net zo als in het vorige onderdeel.

Blz. 38, it. 6: Geen uitwerking.

Blz. 38, it. 7:

$$\begin{aligned}
& x \in f^{-1}[f[A']] \\
\Leftrightarrow\ & f(x) \in f[A'] \\
\Leftrightarrow\ & \text{er is een } x' \in A', \text{ zo dat } f(x') = f(x) \\
\Leftrightarrow\ & \text{er is een } x' \in A', \text{ zo dat } f(x') = f(x)
\end{aligned}$$

dat hoeft niet te betekenen dat x zelf in A' ligt. Dus niet waar. We moeten nog wel een tegenvoorbeeld construeren. Dat gaat nu makkelijk: neem domein $X = \{0, 1\}$, co-domein $Y = \{2\}$, $f = \{(0, 2), (1, 2)\}$, $A' = \{0\}$. Dan

$$A' = \{0\},$$
$$f^{-1}[f[A']] = f^{-1}[f(\{0\})] = f^{-1}[\{2\}] = \{0, 1\}.$$

Blz. 38, it. 8: Waar.

$$\begin{aligned}
& x \in A' \\
\Leftrightarrow\ & \text{er is een } x' \in A', \text{ zo dat } f(x') = f(x) \quad \text{(namelijk } x \text{ zelf!)} \\
\Rightarrow\ & \ldots
\end{aligned}$$

(zie vorige opgave)

$$\Rightarrow \quad x \in f^{-1}[f[A']].$$

Blz. 38, it. 9: Geen uitwerking.

Blz. 38, it. 10: Geen uitwerking.

Blz. 38, it. 11: Waar. Stel $f : X \to Y$ en $g : Y \to Z$. Het makkelijkst is om te beginnen met $z \in g(f[A'])$ willekeurig:

$z \in g(f[A'])$
$\Rightarrow$ er is een $y \in f[A']$, zo dat $g(y) = z$
$\Rightarrow$ er zijn elementen $x \in A'$ en $y \in f[A']$, zo dat $f(x) = y$ en $g(y) = z$
$\Rightarrow$ er is een $x \in A'$, zo dat $g(f(x)) = z$
$\Rightarrow$ er is een $x \in A'$, zo dat $(g \circ f)(x) = z$
$\Rightarrow$ $z \in (g \circ f)[A']$.

Omgekeerd: stel $z \in (g \circ f)[A']$ willekeurig:

$z \in (g \circ f)[A']$
$\Rightarrow$ er is een $x \in A'$, zo dat $(g \circ f)(x) = z$
$\Rightarrow$ er is een $x \in A'$, zo dat $g(f(x)) = z$
$\Rightarrow$ er is een $y \in f[A']$ [neem namelijk $y = f(x)$], zo dat $g(y) = z$
$\Rightarrow$ $z \in g(f[A'])$.

Blz. 38, it. 12: Waar. (Als je 's ochtends eerst je sokken aantrekt en dan je schoenen, dan trek je 's avonds eerst je schoenen uit en dan je sokken .....) Omdat deze identiteit met inverse beelden werkt is het bewijs weer meer rechttoe-rechtaan:

$x \in (g \circ f)^{-1}[C']$
$\Leftrightarrow$ $(g \circ f)(x) \in C'$ [per def. van inverse beeld]
$\Leftrightarrow$ $g(f(x)) \in C'$ [per def. van $g \circ f$]
$\Leftrightarrow$ $f(x) \in g^{-1}[C']$ [per def. van inverse beeld]
$\Leftrightarrow$ $x \in f^{-1}[g^{-1}[C']]$ [per def. van inverse beeld]

Blz. 38, it. 3.21: B moet minstens evenveel elementen tellen als A, dus minstens 15.

Blz. 38, it. 3.22: De functie f is

- *niet surjectief*. Alle oneven getallen vallen buiten het bereik van f.
- *injectief*. Verschillende getallen uit het domein kunnen na verdubbeling nooit hetzelfde getal opleveren.
- *niet bi-jectief*. De functie is niet zowel surjectief als injectief.

Blz. 39, it. 1: Zij $f : A \to B$ een injectie. We gaan een surjectie $g : B \to A$ definiëren.

Zij $B' = f[A]$ het f-beeld van A in B. Omdat f een injectie is, en per definitie alles van $f[A]$ bereikt , is f bi-jectief naar $f[A]$. Elke bi-jectieve functie heeft een inverse (die automatisch bi-jectief is). Laat $g : f[A] \to A$ die inverse zijn. Deze functie is al surjectief (immers: bi-jectie), maar heeft het verkeerde domein. Kies een $a_0 \in A$. Breid g uit naar $g' : B \to A$ als volgt:

$$g'(b) = \begin{cases} g(b) & \text{als } b \in f[A], \\ a_0 & \text{anders.} \end{cases}$$

Omdat g' een uitbreiding van g is, en g surjectief was, is $g' : B \to A$ zeker surjectief.

Blz. 39, it. 2: Zij $f : A \to B$ een surjectie. We gaan een injectie $g : B \to A$ definiëren.

Omdat f een surjectie is, zijn voor alle $b \in B$, de inverse beelden $f^{-1}[b]$ niet leeg en paarsgewijs disjunct. We kunnen dus voor elke $b \in B$ een element $g(b) \in f^{-1}[b]$ kiezen, zo dat alle $g(b)$ verschillend zijn. Dit definieert een injectie $g : B \to A$.

Het voor elke $b \in B$ een element $g(b) \in f^{-1}[b]$ kunnen *kiezen* veronderstelt de geldigheid van het keuzeaxioma (blz. 51).

Blz. 39, it. 1: Zij $a \in A$ willekeurig. Vanwege $g|_A \equiv f$ geldt nu $g(a) = f(a)$. Omdat a ook in B, geldt nu vanwege $F_B \equiv g$ ook $F(a) = g(a)$, dus ook $F(a) = f(a)$. We hebben laten zien dat F beperkt tot A zich hetzelfde gedraagt als f. Daarmee hebben we dus bewezen dat $F|_A \equiv f$.

Blz. 39, it. 2a: $|Y|^{|X|}$.

Blz. 39, it. 2b: Er blijven $m - r$ elementen over om een functiewaarde uit Y met $|Y| = n$ te kiezen. Dus n^{m-r}.

Blz. 40, it. 3: Dat kan niet anders dan de volgende functie zijn:

$$f : A_1 \times \cdots \times A_5 \to B_1 \times \cdots \times B_5 : (x_1, x_2, x_3, x_4, x_5) \mapsto (2 - x_1, 4 - x_2, 6 - x_3, 8 - x_4, 10 - x_5)$$

Signaturen:

$$f_i \circ p_i : A_1 \times \cdots \times A_5 \to B_i : (x_1, x_2, x_3, x_4, x_5) \mapsto x_i \mapsto 2i - x_i$$

en

$$p_i \circ f : A_1 \times \cdots \times A_5 \to B_i : (x_1, x_2, x_3, x_4, x_5) \mapsto (2 - x_1, 4 - x_2, 6 - x_3, 8 - x_4, 10 - x_5) \mapsto 2i - x_i.$$

Blz. 40, it. 3.28: B wordt gekarakteriseerd door de functie f waarbij

$$f = \{\langle a, 0\rangle, \langle b, 0\rangle, \langle c, 1\rangle, \langle d, 1\rangle, \langle e, 1\rangle\}$$

Blz. 40, it. 3.29:

1. f is een functie want voor elke $a \in A$ is er precies één waarde $w \in \{0, 1\}$ zodat $\langle a, w\rangle \in f$: namelijk $w = 1$ als $w \in A'$ en $w = 0$ als $w \notin A'$.
2. het co-domein is inderdaad $\{0, 1\}$.
3. uit de definitie van f volgt $f^{-1}[\{1\}] = A'$.

Blz. 40, it. 1: Stel f is injectief. Kies $a_0 \in A$. Definieer $g : B \to A$

$$g(b) = \begin{cases} a & \text{als } b = f(a) \text{ voor een } a \in A, \\ a_0 & \text{anders.} \end{cases}$$

Kies $a \in A$. Dan per definitie $g(f(a)) = a$.

Stel, omgekeerd, dat $g \circ f = \mathrm{id}_A$. Stel, om te bewijzen dat f injectief is, dat $f(a) = f(a')$. Functie g toepassen op beide kanten levert $g(f(a)) = g(f(a'))$. Omdat $g \circ f = \mathrm{id}_A$ geldt nu $a = a'$, dus f is injectief.

Blz. 40, it. 2: Stel f is surjectief. Omdat f een surjectie is, kunnen we dus voor elke $b \in B$ een element $g(b) \in A$ kiezen, zo dat $f(g(b)) = b$. Dit definieert een functie $g : B \to A$. Kies $b \in B$. Dan per definitie $f(g(b)) = b$.

Stel, omgekeerd, dat $f \circ g = \mathrm{id}_B$. Stel, om te bewijzen dat f surjectief is, dat $b \in B$. Per definitie wordt b door f bereikt middels $g(b)$: $f(g(b)) = b$.

Blz. 41, it. 3: Stel, f is bi-jectief. Dus injectief en surjectief, dus bezit recht- en links inverse g. Deze g is volwaardige inverse (ga na).

Stel, omgekeerd dat f een links- en een rechtsinverse bezit. Dan is f injectief en surjectief, dus bi-jectief.

Blz. 42, it. 3.36: Deze persoon heeft gelijk. Volgens de definitie van partiële functies is elke functie partieel. Totale functies zijn een bijzonder soort partiële functies.

Blz. 42, it. 3.37: Nee, $c = 0$.

Blz. 43, it. 4.1: Zinnen kunnen volgens dit recept weliswaar steeds langer gemaakt worden, maar zullen toch steeds een eindige lengte hebben. De redenering toont dus niet aan dat er *oneindig lange* zinnen bestaan, maar welke dat er zinnen langer dan elke willekeurig lengte gemaakt kunnen worden. (En ook dat er oneindig veel verschillende Nederlandse zinnen zijn.)

Blz. 45, it. 4.3: $f : \mathbb{N} \to \mathbb{Z}$ zodanig dat

$$f(n) = \begin{cases} -\frac{1}{2} \cdot n & \text{als } n \text{ is even} \\ \frac{1}{2} \cdot (n+1) & \text{als } n \text{ is oneven} \end{cases}$$

Blz. 45, it. 4.4: Laat f Cantor's aftelling zijn van de positieve breuken en g een analoge aftelling van de negatieve breuken. Hieruit verkrijgen we een aftelling voor *alle* breuken door bij nul ($= \frac{0}{1}$) te beginnen en vervolgens om en om elementen van f en g te nemen.

Blz. 45, it. 4.5: Laat $\{a_0, a_1, a_2, a_3, \ldots\}$ een aftelbaar alfabet A zijn. Een ordening naar lengte van de verzameling rijtjes over A resulteert niet zonder meer in een aftelling: als we beginnen met alle rijtjes van lengte 1 (een opsomming van A) komen we nooit toe aan rijtjes van 2 tekens ... Ook een 'alfabetische' ordening werkt niet: er zijn al oneindig veel rijtjes die met a_0 beginnen, dus rijtjes die met a_1 aanvangen komen niet aan bod. We kunnen beide ordeningen echter wel vruchtbaar combineren tot een werkende aftelling: begin met a_0, het enige rijtje van lengte 1 waarin het eerste teken van het alfabet voorkomt; vervolg met rijtjes van lengte hoogstens 2 waarin de eerste twee symbolen uit A voorkomen, dat wil zeggen: $a_0a_0, a_0a_1, a_1, a_1a_0, a_1a_1$. Daarna komen de rijtjes met lengte ≤ 3 van tekens met index < 3, en dus in het algemeen: voeg voor elke opvolgende i de rijtjes met lengte $\leq i$ van tekens met index $< i$ aan de aftelling toe. Cruciaal is dat er op de i^{e} stap maar *eindig* veel rijtjes worden toegevoegd (tot een totaal van $\frac{i^{i+1}-i}{i-1}$ rijtjes), en dat toch elk rijtje zo ooit aan bod komt (ga dit na).

Blz. 45, it. 4.6: Alle gasten schuiven een kamer op, zodat de eerste kamer vrij komt voor de extra gast.

Blz. 45, it. 4.7: Laat alle gasten verhuizen. De gast van kamer n gaat naar kamer $2n$. Nu zijn alle oneven kamers vrij voor de buspassagiers.

Blz. 45, it. 4.8: Haal alle gasten uit bed en zet ze in een rij op volgorde van hun kamernummer: $f_0 = 1\ 2\ 3\ 4\ 5\ 6\ \ldots$

Maak een aftelling $f_1, f_2, f_3, \ldots$ van de Hilbert-bussen, en laat elke f_i (voor $i > 0$) een aftelling zijn van passagiers in de desbetreffende bus. Iedereen die onderdak zoekt kan nu als volgt worden opgesteld:

f_0	=	1	2	3	4	5	6	$\cdots$
f_1	=	$f_1(1)$	$f_1(2)$	$f_1(3)$	$f_1(4)$	$f_1(5)$	$f_1(6)$	$\cdots$
f_2	=	$f_2(1)$	$f_2(2)$	$f_2(3)$	$f_2(4)$	$f_2(5)$	$f_2(6)$	$\cdots$
f_3	=	$f_3(1)$	$f_3(2)$	$f_3(3)$	$f_3(4)$	$f_3(5)$	$f_3(6)$	$\cdots$
f_4	=	$f_4(1)$	$f_4(2)$	$f_4(3)$	$f_4(4)$	$f_4(5)$	$f_4(6)$	$\cdots$
f_5	=	$f_5(1)$	$f_5(2)$	$f_5(3)$	$f_5(4)$	$f_5(5)$	$f_5(6)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Cantor's "slingermethode" (vergelijk de aftelinstructie voor de positieve breuken) levert nu aan elke gast een kamer.

Blz. 46, it. 4.9: Stel eerst $A \leq_1 B$. Dan bestaat er dus een injectie van A naar B. Nu kunnen er twee dingen aan de hand zijn:

1. Er bestaat wel omgekeerde injectie van B naar A.
2. Er bestaat geen omgekeerde injectie van B naar A.

In het eerste geval geldt $A =_1 B$; in het tweede geval geldt $A <_1 B$.

Stel omgekeerd dat $A <_1 B$ of $A =_1 B$. Als $A <_1 B$ dan *a fortiori* $A <_1 B$. Als $A =_1 B$ dan bestaat er een bi-jectie tussen A en B dus zeker een injectie van B naar A.

Blz. 46, it. 1: Neem de inbedding $\iota : \mathbb{N} \to \mathbb{Q}$.

Blz. 46, it. 4: Neem de inbedding $\iota : A \to B$.

Blz. 46, it. 4: Elke inbedding $f : A \to B : x \mapsto x$ is een injectie.

Blz. 46, it. 4.12: Dat $\mathbb{N} \times \mathbb{N} =_1 \mathbb{N}$ volgt uit een eerdere opgave (Cantor's zig-zag methode). Dat

$$\mathbb{N} =_1 \mathbb{Z} =_1 \mathbb{Q} =_1 \mathbb{N} \times \mathbb{N} =_1 \mathbb{Z} \times \mathbb{Z} =_1 \mathbb{Q} \times \mathbb{Q}$$

volgt uit $\mathbb{N} =_1 \mathbb{Z} =_1 \mathbb{Q}$ en de implicatie

$$A =_1 B \Rightarrow A \times A =_1 B \times B.$$

We bewijzen het laatste. Stel $A =_1 B$ dan is er een bi-jectie tussen A en B. Maar dan is

$$(a_1, a_2) \mapsto (f(a_1), f(a_2))$$

óók een bi-jectie tussen $A \times A$ en $B \times B$. (Het is niet moeilijk zelf na te gaan deze laatste functie van paren naar paren inderdaad 1-1 en surjectief is.)

Nu nog te bewijzen dat $\mathbb{R} \times \mathbb{R} =_1 \mathbb{R}$. We bewijzen eerst dat $(0,1)^2 =_1 (0,1)$, door coördinaten in het vierkant $(0,1)^2$ decimaal te representeren, en te bewijzen dat

$$(0.d_1d_2d_3\ldots, 0.d'_1d'_2d'_3\ldots) \mapsto 0.d_1d'_1d_2d'_2d_3d'_3\ldots \qquad (10)$$

injectief is. Welnu, wat zou kunnen verhinderen dat (10) niet 1-op-1 is? Als twee verschillende cijferreeksen leiden naar hetzelfde getal. Wanneer is dat? Dat is als twee paren worden afgebeeld op twee verschillende expansies die eigenlijk hetzelfde getal vertegenwoordigen, zoiets als

$$0.1234999\ldots \qquad \text{en} \qquad 0.1235000\ldots.$$

Ga na dan één van de twee paren dan in beide coördinaten een repeterende 9 in de decimale expansie bezit, en het andere paar niet. Als we nu gewoon repeterende 9's in paren verbieden, dan zien we dat de functie (10) nog steeds $(0,1)^2$ als domein heeft, en echt injectief is.

Blz. 46, it. 1: Een compositie van twee injecties is een injectie.

Blz. 47, it. 4.15: De functie $(g^{-1})|_{A_B}$ is welgedefinieerd, maar er is niet zo makkelijk van te bewijzen dat deze surjectief is. Van $g|^{-1}_{B_B}$ gaat dat makkelijker, omdat gebruik kan worden gemaakt van de eigenschappen van $g|_{B_B}$.

Blz. 48, it. 4.16: Stel eerst $A <_1 B$. Per definitie geldt dan $A \leq_1 B$ en $A \neq_1 B$. Uit $A \leq_1 B$ concluderen we dat er een injectie van A naar B bestaat. Als er wel een omgekeerde injectie zou bestaan, dan zou $A =_1 B$, wat niet strookt met onze aanname. Dus er bestaat geen omgekeerde injectie van B naar A.

Stel omgekeerd dat een injectie van A naar B bestaat, maar niet omgekeerd. Uit het feit dat er een injectie van A naar B bestaat kunnen we $A \leq_1 B$ afleiden. Verder kunnen we afleiden dat $A \neq_1 B$ door aan te nemen dat $A =_1 B$. Als $A =_1 B$ zou gelden, dan zou er een bi-jectie tussen A en B bestaan, dus ook een injectie van B naar A, en die conclusie strookt niet met onze oorspronkelijke aanname. We moeten concluderen dat $A \neq_1 B$.

Blz. 48, it. 4.17: Stel $f : A \to B$ is toch een injectie. Per definitie betekent $B <_1 A$ zo veel als

$$B \leq_1 A \qquad (11)$$

maar

$$A \neq_1 B. \qquad (12)$$

Stel $f : A \to B$ is een injectie. Dan per definitie

$$A \leq_1 B. \qquad (13)$$

Met de stelling van Schröder-Bernstein geldt (11), (13) $\Rightarrow$ $A =_1 B$, wat in strijd is met (12). Dus f is niet injectief.

Blz. 48, it. 1: Een aftelling g van $A \cup B$ ontstaat nu door eerst $B - A$ op te sommen en te vervolgen met de aftelling van de A. Als $B - A$ k elementen heeft en f een aftelling van A is, dan kan g formeel als volgt worden gedefinieerd:

$$g(n) = \begin{cases} \text{het } n^e \text{ element van } B - A & \text{als } n < k \\ f(n-k) & \text{als } n \geq k \end{cases}$$

Blz. 48, it. 2: (Vergelijk opdracht 3.6) Laat f een aftelling zijn van A, en g een aftelling van B. Definieer nu een aftelling h van $A \cup B$ als volgt: neem om en om het volgende element van A (volgens aftelling f) dat je nog niet bent tegengekomen en het volgende nieuwe element van B (volgens g).

Blz. 48, it. 3: Er geldt $A \cap B \subseteq A$, en A is aftelbaar. Een deelverzameling van een aftelbare verzameling is aftelbaar.

Blz. 48, it. 4: Laat $a_0, a_1, a_3, \ldots$ een aftelling zijn van A, en $b_0, b_1, b_2, b_3, \ldots$ van B. Dan kunnen de elementen van $A \times B$ weer als volgt worden gerangschikt:

$$\begin{array}{ccccc} \langle a_0, b_0\rangle & \langle a_0, b_1\rangle & \langle a_0, b_2\rangle & \langle a_0, b_3\rangle & \cdots \\ \langle a_1, b_0\rangle & \langle a_1, b_1\rangle & \langle a_1, b_2\rangle & \langle a_1, b_3\rangle & \cdots \\ \langle a_2, b_0\rangle & \langle a_2, b_1\rangle & \langle a_2, b_2\rangle & \langle a_2, b_3\rangle & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{array}$$

Tel weer af als in de breuken-procedure van Cantor.

Blz. 48, it. 5: Aftelling f van A is als volgt om te zetten in een aftelling h van $A - B$: h is gelijk aan f, behalve dat de elementen van $A \cap B$ (dat zijn er immers maar eindig veel) worden overgeslagen.

Blz. 48, it. 4.20: Allemaal door herhaald toepassen van het resultaat van de vorige opgave. Officieel gaat dit met volledige inductie naar het aantal gebruikte verzamelingstheoretische operatoren.

Blz. 48, it. 4.21: Neem $X = \mathbb{R}$ en $A = \emptyset$. Had jij dit tegenvoorbeeld ook? Zijn er eenvoudiger tegenvoorbeelden? Zo ja, waarom? Zo nee waarom niet?

Blz. 48, it. 4.22: Vereniging is verzamelingen achter elkaar zetten, doorsnede is deelverzameling-argument.

Blz. 48, it. 4.23: Vereniging is met Cantor-veegmethode, doorsnede is deelverzameling-argument.

Blz. 49, it. 4.24: Neem $f : \mathbb{N} \to \{0,1\}^{\mathbb{N}}$ met $f(n)(k) = 1$ desda $k = n$. Dan is f een injectie.

Blz. 49, it. 4.26: Beschouw eerst de echte beginstukken van $\mathbb{N}$. Noem $B_0 = \{0\}$, $B_1 = \{0,1\}$, en in het algemeen $B_i = \{0,1,\ldots,i\}$. Hieruit is als volgt een aftelinstructie voor de eindige deelverzamelingen van $\mathbb{N}$ te destilleren: neem eerst de deelverzamelingen van B_0: dat levert $\emptyset$ en kijk vervolgens naar de deelverzamelingen van B_1: neem al die deelverzamelingen op die nog niet in de aftelling aanwezig zijn. Herhaal dit procédé voor elke volgende B_i (voor $i \geq 1$ levert dit 2^i nieuwe deelverzamelingen op.)

$$\begin{array}{lrcl} B_0: & 0 & \longrightarrow & \emptyset \\ & 1 & \longrightarrow & \{0\} \\ B_1: & 2 & \longrightarrow & \{1\} \\ & 3 & \longrightarrow & \{0,1\} \\ B_2: & 4 & \longrightarrow & \{2\} \\ & 5 & \longrightarrow & \{0,2\} \\ & 6 & \longrightarrow & \{1,2\} \\ & 7 & \longrightarrow & \{0,1,2\} \\ & \vdots & & \end{array}$$

Een korte recursieve beschrijving van deze aftelling f is:

- $f(0) = \emptyset$
- $f(n) = f(n-2^k) \cup \{k\}$ als $2^k \leq n < 2^{k+1}$

Blz. 49, it. 4.27: Laat f de aftelling zijn van alle eindige deelverzamelingen van $\mathbb{N}$ uit de vorige opgave. Definieer nu een aftelling g van de co-finiete deelverzamelingen van $\mathbb{N}$ door: $g[X] = f(\mathbb{N} - X)$ voor elke co-finiete X.

Blz. 50, it. 1: Overaftelbaarheid Cartesisch product is te bewijzen middels diagonalisatie-argument op van $\prod_{i=1}^{\infty}\{0,1\}$.

Blz. 50, it. 2: Het Cartesisch product

$$\prod_{i=1}^{\infty} A_i$$

is aftelbaar onder de voorwaarde dat bijna alle (= alle op eindig veel na) A_i leeg zijn of singleton. Dit zijn voldoende voorwaarden voor aftelbaarheid. Of dit ook noodzakelijke voorwaarden zijn is niet nagegaan.

Blz. 50, it. 3: De tabellen zien er hetzelfde uit, maar hebben een verschillende functie. In de eerste tabel staan alle elementen uit de rij $\{A_i\}_{i=1}^{\infty}$ die, *per definitie* aftelbaar is. In de tweede tabel wordt Cantor's diagonaalmethode toegepast. In de tweede tabel doen we dus (tegen beter weten in) een poging alle elementen uit de verzameling $\{0,1\}^{\mathbb{N}}$ af te tellen, wat mislukt.

Blz. 50, it. 1: Vermenigvuldig met het product van alle noemers.

Blz. 50, it. 2: $\mathbb{N}$ is aftelbaar, dus voor iedere n is $\mathbb{N}^n$ aftelbaar, dus $\bigcup_{n=0}^{\infty} NN^n$ is aftelbaar. Ga na dat de laatste verzameling 1-op-1 correspondeert met elementen uit $\mathbb{N}[X]$.

[1] 2018.

Blz. 50, it. 3: Zij $p_1, p_2, p3, \ldots$ een aftelling van $\mathbb{Q}[X]$. Het is (hopelijk) bekend dat iedere n^e-graads veelterm precies n complexe wortels heeft, waarvan een aantal $k \leq n$ reëel is. Dus iedere veelterm p_i bezit in ieder geval eindig veel reële wortels.

De volgende rij is een aftelling van verzameling reële nulpunten van veeltermen over $\mathbb{Q}$:

> alle wortels van p_1, alle wortels van p_2, alle wortels van p_3, ...

Blz. 50, it. 4.31: Definieer stratum

$$\begin{aligned} S_n = \{&\text{alle veeltermen met constanten binnen } [-n,n], \\ &\text{en niet meer dan } n \text{ termen}, \\ &n \text{ factoren per term}, \\ &\text{en } n \text{ variabelen per term}\} \end{aligned}$$

De beperking op factoren impliceert automatisch een beperking op exponenten. Ga na dat S_n eindig is. Vorm $S = \bigcup_{n=1}^{\infty} S_n$. Een aftelbare vereniging van aftelbare verzamelingen is aftelbaar.

Blz. 51, it. 1: Waar, als je aanneemt dat er zich eindig veel mensen op de campus bevinden. (Een juiste aanname.)

Blz. 51, it. 2: Waar, als je aanneemt dat er zich eindig veel mensen op deze wereld bevinden.

Blz. 51, it. 3: Waar, als je aanneemt dat er dat er eindig veel zandkorrels op de wereld liggen.

Blz. 51, it. 4: Onbekend of deze uitspraak waar is. Momenteel[1] is namelijk onbekend of het universum eindig is. Mocht dat zo zijn dan is deze bewering waar. Als het universum aftelbaar oneindig of overaftelbaar oneindig is, dan is de bewering hoogstwaarschijnlijk omwaar,—tenzij kabouter Plop daadwerkelijk aan de wieg heeft gestaan van aftelbaar veel (in het aftelbare geval) of overaftelbaar veel (in het overaftelbare geval) stukjes materie.

Blz. 51, it. 5: Waar: alleen 2 is niet oneven.

Blz. 51, it. 6: Onwaar. Stel van niet. Dan zijn eindig veel priemgetallen niet deelbaar door zeven, en de rest wel. Maar dat kan niet. Wat er zijn oneindig veel priemgetallen dus "de rest" is een oneindige verzameling. Priemgetallen zijn niet deelbaar door zeven, dus alle getallen in de rest zijn niet deelbaar door zeven.

Blz. 51, it. 7: Waar: "slechts" aftelbaar oneindig veel reële getallen zijn geheel.

Blz. 51, it. 8: Waar: "slechts" aftelbaar oneindig veel reële getallen kunnen geschreven worden als een breuk.

Blz. 51, it. 9: Waar: “slechts” aftelbaar oneindig veel reële getallen kunnen een oplossing zijn van een veelterm met rationele coëffciënten. (Zie onderdeel 6 van Opgave 4.30 op blz. 50.)

Blz. 51, it. 10: Waar. De verzameling van oneindige bitstrings is overaftelbaar. (Geloof je dit niet, zet ze dan op een rij en pas Cantor’s diagonaal-argument toe.) De verzameling bitstrings met eindig veel 1-en is echter “slechts” aftelbaar. Eén manier om die verzameling af te tellen is ze te zien als een verzameling eindige bitstrings, waarbij elke bitstring een oneindige staart van nullen bezit. Elke eindige bitstring heeft maar één zo’n staart en de verzameling eindige bitstrings is aftelbaar.

Blz. 53, it. 4.33: De relatie $\leq_1$ op de klasse V van alle verzamelingen is : (A, B en C zijn elementen van V)

- *reflexief:* Voor elke A geldt dat $A \leq_1 A$. De identieke functie op A is immers een injectie van A naar A.
- *anti-symmetrisch:* Als $A \leq_1 B$ en $B \leq_1 A$, dan $A =_1 B$. Dit volgt direct uit Schröder-Bernstein. Een subtiel punt is dat anti-symmetrie eigenlijk een sterkere eis stelt aan $\leq_1$: namelijk *identiteit* ($A = B$ in bovenstaande formule) in plaats van *gelijkmachtigheid* ($A =_1 B$). $\leq_1$ voldoet duidelijk niet aan deze sterkere eis. Omdat $=_1$ hier echter meer passend is dan $=$, zou het beter zijn de definitie van *anti-symmetrie* wat af te zwakken.
- *transitief:* als $A \leq_1 B$ en $B \leq_1 C$ dan $A \leq_1 C$. Immers als f een injectie is van A naar B en g is een injectie van B naar C, dan is $g \circ f$ een injectie van A naar C. We gaan dit laatste voor de goede orde even na. Stel dat $g \circ f(x) = g \circ f(y)$, dus $g(f(x)) = g(f(y))$. Dan $f(x) = f(y)$ want g is injectief. Maar dan $x = y$ want f injectief. □

Blz. 53, it. 4.34: Bijvoorbeeld: ‘even snel lopen als’ en ‘even rijk zijn als’.

Blz. 54, it. 1: 5.

- Voor $\{1\}$ is er één mogelijke relatie: $\{[1]\}$.
- Voor $\{1,2\}$ zijn er twee mogelijke relaties: $\{[1,2]\}$ en $\{[1],[2]\}$.
- Voor $\{1,2,3\}$ zijn er vijf : $\{[1,2,3]\}$, $\{[1,2],[3]\}$, $\{[1,3],[2]\}$, $\{[1],[2,3]\}$, $\{[1],[2],[3]\}$.

Je ziet dat er een patroon ontstaat: de equivalentieklassen voor $\{1,\ldots,n+1\}$ ontstaan uit die van $\{1,\ldots,n\}$ door het element $n+1$ bij één van de klassen uit de vorige generatie te stoppen, óf in een eigen singleton klasse te stoppen.

Blz. 54, it. 2:

$$B_n = \sum_{j=0}^{n-1} \binom{n-1}{j} B_{n-1-j} = \sum_{j=0}^{n-1} \binom{n-1}{n-1-j} B_{n-1-j} = \sum_{j=0}^{n-1} \binom{n-1}{j} B_j.$$

Immers,

$$\binom{n}{n-k} = \binom{n}{k}$$

voor alle n en k.

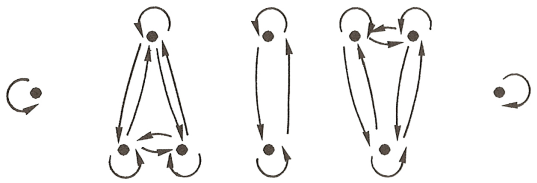
Blz. 54, it. 4.36: –

- $[1]_R = \{x \in N \mid x \text{ is oneven}\} = \{1,3,5,7,\ldots\}$ (de verzameling oneven natuurlijke getallen)
- $[0]_R = \{x \in N \mid x \text{ is even}\} = \{0,2,4,\ldots\}$

Blz. 54, it. 4.37: 121 klassen: voor ieder jaar één, ook voor de baby’s van 0.

Blz. 54, it. 4.38: De pijlrelatie induceert 3 equivalentie-klassen.

Blz. 55, it. 4.39: De equivalentierelatie



Blz. 55, it. 4.40: $[\emptyset]_{=_1}$ is de verzameling met als elementen alle verzamelingen die even groot zijn als $\emptyset$. Dit is alleen $\emptyset$ zelf, dus het aantal elementen is 1. De representant $\emptyset$ bevat nul elementen.

Blz. 55, it. 4.41: $2^{\{\emptyset\}}$ is een representant van $[2^{\{\emptyset\}}]_{=_1}$ en $2^{\{\emptyset\}} = \{\emptyset, \{\emptyset\}\}$. Deze (en daarmee elke andere) representant telt dus 2 elementen.

Blz. 57, it. 4.42: We moeten bewijzen dat $\{\{a\},\{a,b\}\} = \{\{c\},\{c,d\}\} \Leftrightarrow a = c$ en $b = d$. Twee verzamelingen A en B zijn gelijk aan elkaar desda voor elke $x \in A$ geldt dat $x \in B$ en omgekeerd. Laat $A = \{\{a\},\{a,b\}\}$ en $B = \{\{c\},\{c,d\}\}$.

Stel nu eerst dat $A = B$. Dan moet gelden dat $\{a\}$ en $\{a,b\}$ de elementen zijn van B. Er zijn dan twee mogelijkheden:

- $\{a\} = \{c\}$ en $\{a,b\} = \{c,d\}$. Uit het eerste volgt $a = c$ en daarmee uit het tweede ook $b = d$.
- $\{a\} = \{c,d\}$ en $\{a,b\} = \{c\}$. Dit impliceert dat $a = c = d$ en $a = b = c$, kortom $a = b = c = d$.

In beide gevallen geldt dus $a = c$ en $b = d$.

Omgekeerd, als $a = c$ en $b = d$, dan hebben A en B dezelfde elementen, zoals eenvoudig valt na te gaan. Dus dan $A = B$. □

Blz. 57, it. 4.43: Een andere codering is bijvoorbeeld $\langle a,b\rangle = \{\{a,1\},\{b,2\}\}$. Het bewijs is analoog aan het vorige. □

Blz. 61, it. 1: Exponenten van een priemontbinding moeten altijd corresponderen met een rangnummer van een symbool. Dat kan op twee manieren fout gaan: (1) symbolen zijn niet genummerd vanaf 1 of in de nummering zitten gaten; (2) exponenten van een priemontbinding zijn te groot om te kunnen worden terugvertaald naar een symbool. Probleem (1) is op te lossen door alle symbolen doorlopend te nummeren vanaf 0; probleem (2) is op te lossen door grote exponenten te laten verwijzen naar spaties (maar dan heb je geen unieke strings meer) of, als je unieke strings wilt, door geïndexeerde spaties. Door spaties te indexeren, i.e., door er een nummertje aan te hechten ontstaan oneindig veel verschillende spaties.

Blz. 61, it. 2: Afhankelijk van wat je hebt afgesproken:

1. Ongedefinieerd. (Omdat er geen symbolen bij grote getallen horen.)
2. De string "ha" gevolgd door drie spaties: "ha___".
3. De string "ha" gevolgd door drie geïndexeerde spaties:

   "ha$_{9,761,451}$ $_{7,116,167}$ $_{9,761,666}$".

Blz. 62, it. 5.5: Ieder programma dat een berekenbare functie f implementeert, kan worden veranderd in een syntactisch ander programma door er spaties of commentaar aan toe te voegen. We krijgen een andere string en dus een ander programma. De parser zal uiteraard terugkomen met eenzelfde structuur, en de semantiek (betekenis) van het programma zal natuurlijk niet afwijken, maar dat is juist wat gevraagd werd.

Blz. 62, it. 5.6: Zolang een programma nog "bezig" is, i.e., nog wordt uitgevoerd zonder dat er een resultaat is afgedrukt, kunnen we niet zeggen of j bezig is $f(n)$ te berekenen danwel gewoon maar doorloopt zonder ooit nog iets af te drukken. Zolang een programma nog "bezig" is kunnen we dus in het algemeen nooit weten of het ooit zal stoppen.

Blz. 63, it. 2: De constante functie $f : \mathbb{N} \to \mathbb{N} : x \mapsto 0$ is berekenbaar maar niet inverteerbaar. I.h.a. geldt dat niet-injectieve functies niet inverteerbaar zijn.

Blz. 63, it. 1: Programma j heeft input n onmiddellijk als output terug.

Blz. 63, it. 2: Laat j en j' twee programma's zijn die f resp. f' uitrekenen. Als we $f' \circ f(n)$ willen weten, dan laten we eerst j los op n. Als $f \uparrow n$ zal $f' \circ f \uparrow f(n)$ en zal j terecht niet stoppen. Als $f \downarrow n$ zal j uiteindelijk de waarde $f(n)$ teruggeven, waar j' verder mee aan de slag kan. Als $f' \uparrow n$ zal $f' \circ f \uparrow n$ en zal j' terecht niet stoppen. In het andere geval zal j' de waarde $f(f'(n)) = f' \circ f(n)$ teruggeven.

Blz. 63, it. 3: $f : \mathbb{N} \to \mathbb{N}$ is bi-jectief heeft dus een inverse f^{-1}. $f : \mathbb{N} \to \mathbb{N}$ is berekenbaar en injectief. Met het antwoord van Opgave 5.5 (blz. 62) kunnen we dus concluderen dat f^{-1} berekenbaar is. Als $g \circ f$ berekenbaar zou zijn, dan zou $(g \circ f) \circ f^{-1}$ dat ook zijn. (De compositie van twee berekenbare functies is berekenbaar.) Maar

$$(g \circ f) \circ f^{-1} = g \circ (f \circ f^{-1}) = g \circ I = g.$$

Maar g is onberekenbaar. Tegenspraak. (Merk op dat $f \circ f^{-1}$ echt I is, want f^{-1} is totaal en bi-jectief.)

Blz. 63, it. 1: Stel X is beslisbaar. Er is dus een programma $j/1 \in J$ dat over X kan beslissen. Per definitie is de karakteristieke functie dan berekenbaar! Omgekeerd is de implicatie net zo direct: stel dat de karakteristieke functie van X berekenbaar. Er is dus een programma, $j/1$, welke $\chi(X)$ kan uitrekenen. Precies dit programma beslist over lidmaatschap in X!

Blz. 63, it. 2: Schrijf een (mogelijk gigantisch maar altijd eindig) case-statement die lidmaatschap in X verifieert.

Blz. 63, it. 3: Stel dat j_X over X kan beslissen, en j_Y over Y. Zij $n \in \mathbb{N}$ geven. Voor $j_X(n)$ en $j_Y(n)$ na elkaar uit. Dat kan, want zowel $j_X(n)$ als $j_Y(n)$ zijn beslissingsalgoritmen. Combineer na afloop beide antwoorden.

Blz. 63, it. 5.11: De oneindige rij

$$(1+\frac{1}{2})^2, (1+\frac{1}{3})^3, (1+\frac{1}{4})^4, \ldots$$

stijgt naar e, en de oneindige rij

$$\frac{1}{(1-\frac{1}{2})^2}, \frac{1}{(1-\frac{1}{3})^3}, \frac{1}{(1-\frac{1}{4})^4}, \ldots$$

daalt naar e. (Sneller dan dat de eerste rij naar e stijgt, maar dat is verder irrelevant.)

Een programma om te kunnen beslissen over $E = \{q \in \mathbb{Q} \mid q < e\}$ zou als volgt kunnen werken. Laat $x \in \mathbb{Q}$ het getal zijn waarvoor we moeten beslissen of het links of rechts van e ligt. (Er geldt $x \neq e$ omdat $e \notin \mathbb{Q}$). Ontwikkel nu steeds om beurten een element uit één van beide reeksen. Overschrijdt de stijgende reeks op enig moment x, dan was $x < e$. Daalt de andere reeks op enig moment onder x, dan was $e < x$. Omdat $x \neq e$ moet één van beide gebeurtenissen ooit plaatsvinden.

Blz. 63, it. 5.12: A problem is decidable if there exists a computer program that decides it. However, we don't need to be abe to construct one program. We simply have to prove the existence of a program that decides it. Pick a program M1 that always prints YES and halts, and pick another program M2 that always prints NO and halts. We have two cases: If aliens exist then program M1 does the job, if aliens don't exist then program M2 does the job. So in both cases there is a program that does the job and this problem is hence decidable (of course we don't know which of these two programs actually decides it, but we know that such a program exists!).

In general, for any one question, or any finite number of questions, there is always a decider. Decidability becomes interesting only when the language is infinite. So,

the set of all theorems mankind has proven is, trivially, decidable. This does not mean that we are over-trivializing unknown problems. It just means that the theory of computation considers any finite language trivial. The theory of computation is more about the structure of infinite sets (languages) that can be defined using finite means (like a machine). What these sets mean is not, technically speaking, part of the study. This is often a source of great confusion in people who don't under- stand this difference, but use Turing's undecidability arguments in their philosophical arguments (very similar to how quantum theory is misinterpreted in many arguments).

Voor de Goldbach-functie geldt hetzelfde verhaal. De functie is berekenbaar, maar we weten niet hoe de functie er uit ziet, meer bepaald of deze identiek 1 of identiek 0 is.

Blz. 63, it. 5.13: Veronderstel eerst dat f een totale niet-dalende berekenbare functie is. Als f's bereik eindig is, dan is f's bereik zeker beslisbaar. (Maak zo nodig Opgave 2, blz. 63.) Als f's bereik oneindig is, dan kunnen we er als volgt over beslissen. Stel $x \in N$. Enumereer f totdat $f(n) = x$ (print 1) of $f(n) > x$ (print 0, immers f is niet-dalend). Omdat f een oneindig bereik heft moet één van de twee gebeurtenissen zich uiteindelijk voordoen.

Veronderstel omgekeerd dat X beslisbaar is. Definieer f door $f(0) = 0$ en

$$f(n) = \begin{cases} n, & n \in X \\ f(n-1), & \text{anders.} \end{cases}$$

Het bereik van f is nu $X \cup \{0\}$ (ga na!). De functie f is nu makkelijk om te zetten naar een functie f' met bereik X, door het eerste stuk (met bereik $\{0\}$) te liften naar het minimum van X. (Teken de grafiek in geval van onduidelijkheid!) Om aan te tonen dat f niet-dalend is, toen we eerst dat $f(n) \leq n$, voor alle n. Dit kan worden gedaan met volledige inductie naar n. Nu $f(n) = n > n-1 \geq f(n-1)$ danwel $f(n) = f(n-1)$. Dus in beide gevallen $f(n) \geq f(n-1)$. De volgende pseudo-code toont dat f berekenbaar is:

Gegeven: n. $r := n$.
zolang $\chi(X)(r) = 0$ en $r > 0$ **doe** {
    $r := r - 1$
}
Geef r terug.

Blz. 64, it. 5.14: De verzameling X bevat in ieder geval de eerste 34 kwadraten en mogelijk meer. Het programma j kan ergens na die twee uur nog getallen afdrukken (dan bevat X meer), of niet (en dan is X eindig). Dat is alles wat we na twee uur weten.

Blz. 64, it. 1: Stel $j/0$ somt X op. Voeg aan een eind van j een oneindige while-loop o.i.d. toe, zodat we zeker weten dat j nooit zal stoppen. (Wat dat voor zin heeft zien we later.) Het volgende programma, $j'/1$, stopt precies op alle elementen van X:

> Stel n is gegeven. Voer j uit, en wacht totdat n wordt afgedrukt.
> – Als $n \in X$ zal j dit getal afdrukken. Programma j' kan dan j onderbreken, en daarna zelf ook stoppen. stoppen.
> – Als $n \notin X$ zal j dit getal nooit afdrukken, en zal j, dus ook j', nooit stoppen.

In de laatste conditie gebruikten we de kennis dat j nooit zal stoppen.

Stel, omgekeerd, dat $j/1$ een programma is dat precies op alle elementen van X stopt. Het volgende programma $j'/0$ somt X op:

> Voer j "quasi-parallel" uit op $\mathbb{N}$:
>
> - eerst één stap van $j(0)$
> - dan één stap van $j(0)$ en één stap van $j(1)$ (de uitvoering van $j(0)$ is nu twee stappen ver)
> - dan één stap van $j(0)$, één stap van $j(1)$, en één stap van $j(2)$ (de uitvoering van $j(0)$ is nu drie stappen ver, de uitvoering van $j(1)$ is twee stappen ver, en de uitvoering van $j(2)$ is één stap ver)
> - ...
>
> Als nu alle n waarvoor $j(n)$ eindigt worden afgedrukt, dan krijgen we op deze manier een opsomming van X.

Blz. 64, it. 2: X heet *semi-beslisbaar* als en slechts als er een programma $j/1 \in J$ bestaat dat stopt **en** 1 **afdrukt** op precies alle elementen uit X. Uiteraard zijn programma's die stoppen op input n om te schrijven naar programma's die stoppen op input n en daarna een 1 af drukken, en omgekeerd.

Blz. 65, it. 5.16: Veronderstel dat een niet-lege verzameling $X \subseteq \mathbb{N}$ opgesomd wordt door j. Laat $x_0 \in X$ willekeurig. Beschouw de volgende totale functie f:

$$f(n) = \begin{cases} x & \text{als } f \text{ na } n \text{ stappen resultaat } x \text{ geeft,} \\ x_0 & \text{anders.} \end{cases}$$

(We nemen aan dat per rekenstap hoogstens één getal kan worden afgedrukt. Is dat niet zo, dan verkleinen we de stapgrootte.)

Blz. 65, it. 5.17: De eerste methode is correct maar de tweede methode niet. De eerste methode is correct omdat het altijd mogelijk is de uitvoering van twee programma's te vervlechten door eerst de één wat rekentijd te geven, daarna de ander wat rekentijd te geven, etc. De tweede methode is niet correct omdat het bij opsom-programma's geen zin heeft te gaan wachten op het volgende getal dat wordt afgedrukt, omdat zo'n volgende getal misschien niet meer komt. Herinner dat een opsom-programma niet verplicht is te stoppen na het afdrukken van een eventueel laatste getal.

Blz. 65, it. 1: Zij j en j' programma's die X resp. Y opsommen. Door de uitvoering van j en j' af te wisselen, ontstaat een opsomming van $X \cup Y$. (Herinner dat het afdrukken van doublures niet verboden is.)

Blz. 65, it. 2: Laat j en j' hun uitvoer niet afdrukken, maar wegstoppen in twee datastructuren D resp. D'. Wissel de uitvoering van j en j' af. Stopt j een volgend getal n in D, controleer dan of $n \in D'$, zo ja druk n af. Stopt j' een volgend getal n in D', controleer dan of $n \in D$, zo ja druk n af.

Blz. 65, it. 5.19: Laat $j/0$ een non-deterministisch programma zijn dat zo af en toe getallen afdrukt. we voeren de volgende notatie in:

$$(0, n_1, \ldots, n_k)$$

correspondeert met de situatie dat j inmiddels k keuzes heeft gemaakt met uitkomst $n_1, \ldots, n_k$. Dus j wordt vertegenwoordigt door (0), en f na te hebben gekozen voor 5 bij de eerste keuzemogelijkheid, 3 bij de tweede keuzemogelijkheid en 999 bij de derde keuzemogelijkheid, wordt vertegenwoordigt door $(0,5,3,999)$. Laat s het aantal stappen zijn sinds de laatste keuzemogelijkheid. Omdat

$$Z = \{(0, n_1, \ldots, n_k, s) \mid k \geq 0, n_i \geq 0, 1 \leq i \leq k, s \geq 0\}$$

aftelbaar is (enumereer bijvoorbeeld door max.{max. length, max. element, max. steps}), alle executie-takken kunnen worden gesimuleerd door een deterministisch programma dat één voor één de elementen van Z afdrukt.

De andere kant van de equivalentie is makkelijk, omdat ieder deterministisch programma per definitie een non-deterministisch programma is. (Een non-deterministisch programma is immers een programma waar op bepaalde punten verschillende het berekeningstraject *kan* vertakken.)

Blz. 65, it. 5.20: Laat $j/0$ de verzameling X opsommen, en laat $j'/0$ de verzameling Y opsommen. Voer j en j' pseudo-gelijktijdig uit door hun uitvoering te vervlechten. Drukt j een getal m af, druk dan alle paren

$$\{(m, y) \mid y \text{ reeds afgedrukt door } j'\}$$

af. Drukt j' een getal n af, druk dan alle paren

$$\{(x, n) \mid x \text{ reeds afgedrukt door } j\}$$

af. Op deze manier wordt $X \times Y$ opgesomd.

Blz. 65, it. 5.21: Allemaal door herhaald toepassen van het resultaat van Opgave 5.18 en 5.20. Officieel gaat dit met volledige inductie naar het aantal gebruikte verzamelingstheoretische operatoren.

Blz. 65, it. 1: Laat X_i opgesomd worden door programma j_i, waarbij $i \geq 0$. Voer de programma's $\{j_i\}_{i=0}^{\infty}$ incrementeel quasi-parallel uit:

- eerst één stap van j_0
- dan één stap van j_0 en één stap van j_1 (de uitvoering van j_0 is nu twee stappen ver)
- dan één stap van j_0, één stap van j_1, en één stap van j_2 (de uitvoering van j_0 is nu drie stappen ver, de uitvoering van j_1 is twee stappen ver, en de uitvoering van j_2 is één stap ver)
- ...

Uiteindelijk wordt alles uit $\cup_{i=1}^{\infty} X_i$ afgedrukt.

Blz. 65, it. 2: In Opgave 4.23, blz. 49 zagen we al dat het aftelbaar oneindige product van eindige verzamelingen helaas niet meer aftelbaar is. Dus zeker niet opsombaar. (Immers, niet aftelbaar impliceert niet opsombaar.)

Blz. 65, it. 5.23: Vermoedelijk niet. Wij kunnen i.i.g. niet bewijzen dat de aftelbare doorsnijding $\cap_{i=1}^{\infty} X_i$ opsombaar is. Het is ook niet erg waarschijnlijk: ook met dovetailing moet een getal bij de opsomming van *alle* instanties zijn verschenen alvorens het afgedrukt mag worden.

Blz. 66, it. 5.24: – Iedere beslisbare verzameling is opsombaar: zij X beslisbaar. Vanwege beslisbaarheid is er een programma j die voor iedere $n \in \mathbb{N}$ kan bepalen of $n \in X$. Schrijf nu het volgende programma j' om X op te sommen:

> Laat j voor elk van de getallen $n = 0, 1, 2 \ldots$ achtereenvolgens bepalen of $n \in X$. Zo ja, druk dan n af. Op deze manier wordt X opgesomd.

– Als een verzameling zowel opsombaar als complementair opsombaar is, dan is die verzameling beslisbaar: laat j een programma zijn dat X opsomt, en laat j' een programma zijn dat $\mathbb{N} \backslash X$ opsomt. Schrijf nu het volgende programma k om te beslissen of $n \in X$:

> Laat n gegeven zijn. Voer j en j' pseudo-gelijktijdig uit door hun executie te vervlechten. Als j het getal n afdrukt, concludeer $n \in X$, en stop. Als j' het getal n afdrukt, concludeer dan dat $n \notin X$, en stop. Omdat
>
> $$n \in \mathbb{N} = X \cup (\mathbb{N} \backslash X)$$
>
> kunnen we er zeker van zijn dat één van beide programma's uiteindelijk n zal afdrukken.

Blz. 67, it. 5.25: Zij j een programma dat over X kan beslissen. Schrijf een programma j' als volgt

> Laat n gegeven zijn. Check of n even is. Als n even is, deel het door twee, en laat j controleren of $n/2 \in X$. Zo ja, laat j' dan 1 teruggeven, zo nee, laat j' dan 0 teruggeven. Als n oneven is, dan kan het zeker geen tweevoud zijn van een element in X, laat j' dan 0 teruggeven.

Blz. 67, it. 5.26: De uitwerking van deze opgave lijkt op de uitwerking van Opgave 5.18. Laat $j/0$ de verzameling

X opsommen, en laat $j'/0$ de verzameling Y opsommen. Het programma j'' dat twee disjuncte verzamelingen X' en Y' afdrukt, zó dat

$$X' \subseteq X,\ Y' \subseteq Y,\ \text{en } X' \cup Y' = X \cup Y,$$

ziet er als volgt uit:

> Voer j en j' pseudo-gelijktijdig uit door hun uitvoering te vervlechten met de zwaluwstaart-methode. Houd ook weer in twee datastructuren D en D' bij wat j en j' afdrukken. Als j op het punt staat een getal n af te drukken, controleren we eerst of het al niet door j' is afgedrukt. Is dat het geval, dan wordt nu niets afgedrukt. Als j' op het punt staat een getal n af te drukken, controleren we eerst of het al niet door j is afgedrukt. Is dat het geval, dan wordt er weer niets afgedrukt. Op deze manier drukken j en j' twee disjuncte verzamelingen X' en Y' af die aan de gewenste eisen voldoen.

Blz. 67, it. 5.27: Zij X een oneindige opsombare verzameling. Per definitie van opsomming bestaat er dus een programma $\pi/0$ welke X kan opsommen. Als deze opsomming geen doublures bevat, zijn we klaar, immers de functie die elk natuurlijk getal n afbeeldt op het n-de opgesomde getal is een totale berekenbare injectieve functie met beeld X (ga na!). (Voor de volledigheid: een implementatie van f, het programma $j/1$, zou als volgt kunnen werken. Het ontvangt een getal, n, en laat π achtereenvolgens n keer opsommen. Het n-de getal is dan het beeld van f.)

Als π's opsomming wel doublures bevat, dan valt er wel een programma $\pi'/1$ te bedenken dat X opsomt zonder doublures (ga na!). Dit programma $\pi'/1$ implementeert dan weer een totale berekenbare injectieve functie met beeld X.

Blz. 67, it. 1: Neem een Diofantische vergelijking $f(n,x_1,\ldots,x_k)=0$. Nu maken we een algoritme dat gewoon alle mogelijke waarden probeert voor n, $x_1,\ldots,x_k$, in stijgende volgorde van de som van hun absolute waarden, en printen n elke keer als $f(n,x_1,\ldots,x_k)=0$. Dit algoritme zal uiteraard altijd lopen en precies een lijst van de n waarvoor $f(n,x1,\ldots,xk)=0$ heeft een oplossing in $x_1,\ldots,x_k$.

Blz. 75, it. 6.1: Bijvoorbeeld de volgende redenering:

> Als ik slaap, studeer ik niet.
> ---
> Als ik niet slaap, studeer ik wel.

Deze redenering is ongeldig, want ik kan wakker zijn en toch niet studeren.

Blz. 75, it. 6.2: Een tegenvoorbeeld is bijvoorbeeld

> Als ik mijn planten geen water geef, gaan ze dood.
> Mijn planten gaan dood.
> ---
> Ik geef mijn planten geen water.

Deze redenering is ongeldig, want planten kunnen ook doodgaan door te weinig zonlicht of door te veel water.

Blz. 78, it. 7.1:

a	1	0
1	0	1
0	1	0

Blz. 79, it. 7.2: $\wedge$, $\vee$, a en $\leftrightarrow$ zijn commutatief. $\rightarrow$ is niet commutatief.

Blz. 79, it. 7.3: We geven de waarheidstafel bij wijze van uitzondering zowel in uitgebreide vorm (links) als in de verkorte (rechts).

p	q	$(p \leftrightarrow q)$	$\neg(p \leftrightarrow q)$
1	1	1	0
0	1	0	1
1	0	0	1
0	0	1	0

$\neg$	$(p$	$\leftrightarrow$	$q)$
0	1	1	1
1	0	0	1
1	1	0	0
0	0	1	0

Bijvoorbeeld $(p \vee q) \wedge \neg(p \wedge q)$ en $p \leftrightarrow \neg q$ zijn logisch equivalent met de gegeven formule.

Blz. 80, it. 7.4: De grammatica genereert 72 zinnen. Immers, Z wordt herschreven tot het rijtje $C\ D\ E\ C\ D$ **.** , zodat er $3 \times 2 \times 2 \times 3 \times 2 = 72$ zinnen zijn.

Blz. 80, it. 7.5: De regel Z ::= **a** Z kan willekeurig vaak worden toegepast, resulterend in het rijtje **a a a . . . a** Z waaruit met de tweede regel **a a a . . . a b** kan worden afgeleid. Er zijn aldus oneindig veel zinnen afleidbaar.

Algemener: een (BNF) herschrijfgrammatica produceert een oneindig aantal zinnen als er uit het startsymbool (Z) een indirecte afleiding (van 0 of meer stappen, notatie $\overset{*}{\Rightarrow}$) bestaat naar een rijtje met een zg. **recursief** of **nestend** element. Preciezer: (u, v, x, y en z zijn willekeurige rijtjes eindsymbolen, u en v niet beide leeg)

- $Z \overset{*}{\Rightarrow} x\,A\,y$ (met als randgeval $A = Z$),
- $A \overset{+}{\Rightarrow} u\,A\,v$, dat wil zeggen A levert in één of meer stappen een rijtje op waar A weer deel van uitmaakt (dit maakt het symbool A en daarmee de grammatica 'nestend')
- $A \overset{+}{\Rightarrow} z$, dus A kan ook tot een eindrijtje worden herschreven.

Blz. 82, it. 7.6: – Basisstap: de propositieletter p begint met een p.

– Inductiestap: als A een propositieletter is en A begint met een p, dan begint A' ook met een p. □

Blz. 82, it. 7.7: **Bewijs:** (met inductie naar de grootte van de verzameling A)

– Basisstap: als A nul elementen heeft, dat wil zeggen $A = \emptyset$, dan heeft $2^(A) = 2^(\emptyset) = \{\emptyset\}$ dus $1 = 2^0$ element.

– Inductiestap: stel dat voor elke verzameling A met $n-1$ elementen geldt dat $2^{(}A)$ 2^{n-1} elementen bevat (de zg. *inductiehypothese*).
Neem nu een verzameling A met n elementen. Kies een vast element $a \in A$. Elke deelverzameling van A bevat a wel of niet. $2^{(}A)$ bevat dus alle deelverzamelingen van $A-\{a\}$ plus al deze deelverzamelingen uitgebreid met a. In formule:

$$2^{(}A) = 2^{(}A-\{a\}) \cup \{X \cup \{a\} \mid X \in 2^{(}A-\{a\})\}$$

Het aantal elementen van $2^{(}A)$ is dus $2 \times 2^{n-1} = 2^n$. □

Blz. 82, it. 7.8:

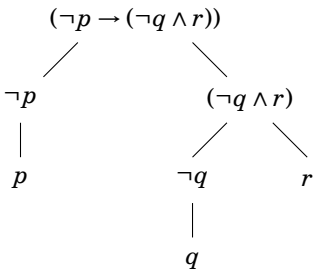
1.	onwelgevormd	8.	onwelgevormd
2.	onwelgevormd	9.	onwelgevormd
3.	welgevormd	10.	welgevormd
4.	welgevormd	11.	onwelgevormd
5.	onwelgevormd	12.	welgevormd
6.	welgevormd	13.	onwelgevormd
7.	onwelgevormd	14.	welgevormd

N.B. Bij 2, 9 en 13 mag blijkens het vervolg van § 7.4 'welgevormd' staan.

Blz. 82, it. 7.9: De verzameling welgevormde formules van $\mathscr{T}$ is *aftelbaar*. Een manier om dat in te zien is de volgende. De verzameling is oneindig omdat alle formules van de vorm $\neg\neg\ldots\neg p$ er toe behoren. Verder is ze een deelverzameling van de verzameling van alle rijtjes over het eindige alfabet $\{p,q,r,\neg,),(,\vee,\wedge,\rightarrow,\leftrightarrow\}$. Zoals we weten uit § 4.2 is deze verzameling tekenrijtjes aftelbaar. Dus moet de verzameling welgevormde formules van $\mathscr{T}$ ook aftelbaar zijn.

Blz. 83, it. 7.10: De verzameling welgevormde formules van deze $\mathscr{T}$ is eveneens aftelbaar. Het bewijs hiervan is volkomen analoog aan het vorige. Cruciaal hierbij is dat in § 4.2 (opdracht 4.5) is bewezen dat ook bij een *aftelbaar* alfabet de verzameling tekenrijtjes aftelbaar is.

Blz. 83, it. 7.11: Het bereik van het eerste negatie-teken is $(p \wedge (q \vee \neg r))$. Het bereik van het tweede negatie-teken is r.



Blz. 83, it. 7.12:

Het bereik van het eerste negatie-teken is p en van het tweede q.

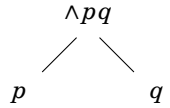
Blz. 83, it. 7.13:

naaktloper ::=
propositieletter | ¬ *formule* |
formule ∧ *formule* | *formule* ∨ *formule* |
formule → *formule* |
formule ↔ *formule*

formule ::=
propositieletter | ¬ *formule* |
(formule ∧ *formule)* | *(formule* ∨ *formule)* |
(formule → *formule)* | *(formule* ↔ *formule)*

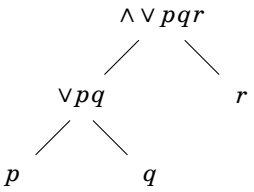
Blz. 84, it. 7.14: – p is identiek aan zijn constructieboom en is al in infixnotatie.

– $\wedge pq$ heeft als constructieboom



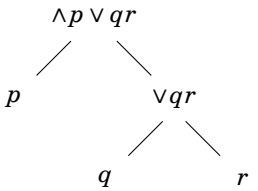
Deze formule correspondeert met $p \wedge q$.

– $\wedge \vee pqr$ heeft als constructieboom



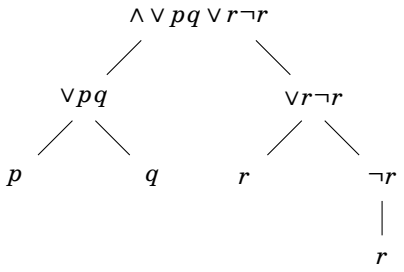
Deze formule correspondeert met $(p \vee q) \wedge r$.

– $\wedge p \vee qr$ heeft als constructieboom



Deze formule correspondeert met $p \wedge (q \vee r)$.

– $\wedge \vee pq \vee r\neg r$ heeft als constructieboom



Deze formule correspondeert met $(p \vee q) \wedge (r \vee \neg r)$.

Blz. 84, it. 7.15: – elke propositieletter uit P is een welgevormde formule van $\mathscr{T}$;

– als φ een welgevormde formule van $\mathscr{T}$ is, dan $\neg\varphi$ ook; als φ en ψ welgevormde formules van $\mathscr{T}$ zijn, dan $\wedge\varphi\psi$, $\vee\varphi\psi$, $\rightarrow\varphi\psi$ en $\leftrightarrow\varphi\psi$ ook;

– niets anders is een welgevormde formule van $\mathscr{T}$.

Blz. 84, it. 7.16:

formule ::=
propositieletter | ¬ *formule* |
∧ *formule formule* | ∨ *formule formule* |
→ *formule formule* | ↔ *formule formule*

Blz. 84, it. 7.17: 1. p 4. $pqr\vee\wedge$

2. $pq\wedge$ 5. $pq\vee rr\neg\vee\wedge$

3. $pq\vee r\wedge$ 6. $r\neg\neg$

Blz. 84, it. 7.18: – elk element van P is een welgevormde formule van $\mathscr{T}$;

– als φ en ψ welgevormde formules van $\mathscr{T}$ zijn, dan $\varphi\neg$, $\varphi\psi\wedge$, $\varphi\psi\vee$, $\varphi\psi\rightarrow$ en $\varphi\psi\leftrightarrow$ ook;

– niets anders is een welgevormde formule van $\mathscr{T}$.

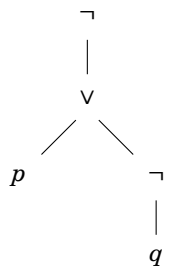
Blz. 84, it. 7.19: – als $p\in P$ dan is (p) een (atomaire) welgevormde formule van $\mathscr{T}$;

– als φ en ψ propositieletters of niet-atomaire welgevormde formules van $\mathscr{T}$ zijn, dan zijn $(\neg\varphi)$, $(\wedge\varphi\psi)$, $(\vee\varphi\psi)$, $(\rightarrow\varphi\psi)$ en $(\leftrightarrow\varphi\psi)$ ook welgevormde formules;

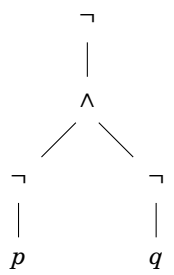
– niets anders is een welgevormde formule van $\mathscr{T}$.

Blz. 84, it. 7.20: 1. de structuurboom van $\neg\neg\neg p$ is:

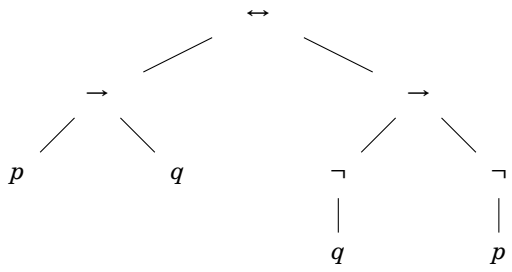
2. de structuurboom van $\neg(p\vee\neg q)$ is:



3. de structuurboom van $\neg(\neg p\wedge\neg q)$ is:



4. de structuurboom van $((p\rightarrow q)\leftrightarrow(\neg q\rightarrow\neg p))$ is:



Blz. 86, it. 8.1:

¬	$(p$	∧	$q)$
0	1	1	1
1	0	0	1
1	1	0	0
1	0	0	0

$(\neg$	p	∨	¬	$q)$
0	1	0	0	1
1	0	1	0	1
0	1	1	1	0
1	0	1	1	0

De formules hebben dezelfde waarheidswaarde voor elke interpretatie van de proposities p en q. Op hoofdkolommen zijn de waarheidstafels immers gelijk.

Blz. 86, it. 8.2:

1.

φ	∨	¬	φ
1	1	0	1
0	1	1	0

2.

$(\varphi$	∧	$\psi)$	→	φ
1	1	1	1	1
0	0	1	1	0
1	0	0	1	1
0	0	0	1	0

3.

φ	→	$(\varphi$	∨	$\psi)$
1	1	1	1	1
0	1	0	1	1
1	1	1	1	0
0	1	0	0	0

4.

¬	φ	→	$(\varphi$	→	$\psi)$
0	1	1	1	1	1
1	0	1	0	1	1
0	1	1	1	0	0
1	0	1	0	1	0

5.

$((\varphi$	→	$\psi)$	→	$\varphi)$	→	φ
1	1	1	1	1	1	1
0	1	1	0	0	1	0
1	0	0	1	1	1	1
0	1	0	0	0	1	0

6.

$(\varphi$	→	$(\psi$	→	$\chi))$	→	$((\varphi$	→	$\psi)$	→	$(\varphi$	→	$\chi))$
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	0	1	1
1	1	0	1	1	1	1	0	0	1	1	1	1
0	1	0	1	1	1	0	1	0	1	0	1	1
1	0	1	0	0	1	1	1	1	0	1	0	0
0	1	1	0	0	1	0	1	1	1	0	1	0
1	1	0	1	0	1	1	0	0	1	1	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0

Blz. 87, it. 8.3: De formule $\neg(p \to p)$ is de enige contradictie. Dit blijkt uit de bijbehorende waarheidstabel die laat zien dat de waarde van de gehele formule steeds 0 is, ongeacht de waarde van de propositieletter:

$\neg$	$(p$	$\to$	$p)$
0	1	1	1
0	0	1	0

Alle andere formules zijn tautologieën. Bij wijze van voorbeeld maken we een waarheidstafel voor 4.)

$(p$	$\wedge$	$q)$	$\leftrightarrow$	$\neg$	$(\neg$	p	$\vee$	$\neg$	$q)$
1	1	1	1	1	0	1	0	0	1
0	0	1	1	0	1	0	1	0	1
1	0	0	1	0	0	1	1	1	0
0	0	0	1	0	1	0	1	1	0

Blz. 87, it. 8.4: Als φ een tautologie is, dan geldt voor elke waardering V dat $V(\varphi) = 1$. Voor elke waardering V geldt nu ook dat $V(\neg\varphi) = 0$; $\neg\varphi$ is dus een contradictie.

Omgekeerd, als φ een contradictie is, dan geldt voor elke waardering V dat $V(\varphi) = 0$, dus ook dat $V(\neg\varphi) = 1$; $\neg\varphi$ is dan een tautologie. □

Blz. 87, it. 8.5: Bijvoorbeeld: p of $p \to q$.

Blz. 88, it. 8.6: Alleen in rijen waar beide premissen waar zijn (aangegeven door 1) moet de waarde van de conclusie onderzocht worden: deze moet dan steeds 1 zijn, wil de redenering geldig zijn.

φ	φ	$\to$	ψ	ψ
1	1	1	1	1
0	0	1	1	1
1	1	0	0	0
0	0	1	0	0

Blz. 88, it. 8.7:

φ	$\to$	ψ	$\neg$	ψ	$\neg$	φ
1	1	1	0	1	0	1
0	1	1	0	1	1	0
1	0	0	1	0	0	1
0	1	0	1	0	1	0

Blz. 88, it. 8.8: 1.

φ	$\to$	ψ	ψ	$\to$	χ	φ	$\to$	χ
1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1
0	1	0	0	1	1	0	1	1
1	1	1	1	0	0	1	0	0
0	1	1	1	0	0	0	1	0
1	0	0	0	1	0	1	0	0
0	1	0	0	1	0	0	1	0

2.

φ	$\vee$	ψ	φ	$\to$	χ	χ	$\vee$	ψ
1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1
1	1	0	1	1	1	1	1	0
0	0	0	0	1	1	1	1	0
1	1	1	1	0	0	0	1	1
0	1	1	0	1	0	0	1	1
1	1	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0

Blz. 88, it. 8.9: 1.

φ	$\neg$	$\neg$	φ
1	1	0	1
0	0	1	0

2.

$\neg$	φ	φ	$\to$	$(\psi$	$\wedge$	$\neg$	$\psi)$
0	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1
0	1	1	0	0	0	1	0
1	0	0	1	0	0	1	0

3.

$\neg$	$(\varphi$	$\wedge$	$\psi)$	$\neg$	φ	$\vee$	$\neg$	ψ
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	1	0	1
1	1	0	0	0	1	1	1	0
1	0	0	0	1	0	1	1	0

4.

$\neg$	$(\varphi$	$\vee$	$\psi)$	$\neg$	φ	$\wedge$	$\neg$	ψ
0	1	1	1	0	1	0	0	1
0	0	1	1	1	0	0	0	1
0	1	1	0	0	1	0	1	0
1	0	0	0	1	0	1	1	0

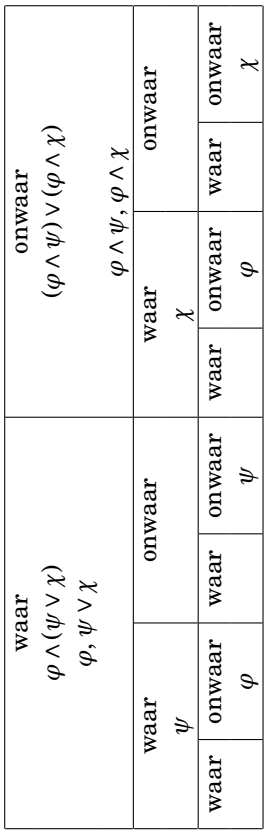
Blz. 88, it. 8.10: Definities uitschrijven.

Blz. 89, it. 1: Ja. Een kortere DNF is bijvoorbeeld $\neg p \vee (p \wedge \neg q)$.

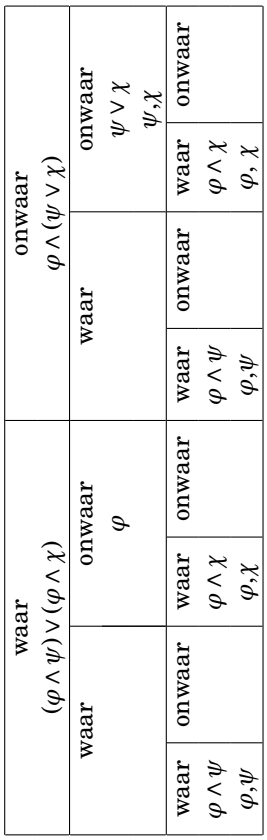
Blz. 89, it. 2: Nee. Zie verder het antwoord van Opgave 8.23.

Blz. 90, it. 8b: Suppose a CNF C is equivalent to (7). Suppose further that there is a clause is which proposition letter p does not occur. This is impossible, for choose a model m such that $m \not\models p$. Flip valuation of p. Clause is still false, while (7) becomes true. If all proposition letter must occur in every clause, then it is easy to see that there must be as many clauses as there are rows in the corresponding truth table with a 1. Hence, C must be exponentially large.

Blz. 93, it. 8.14: 1. Dat $\varphi \wedge (\psi \vee \chi) \models (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$ blijkt uit het volgende tableau:

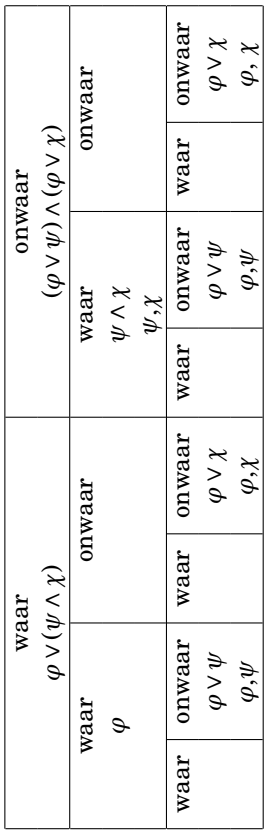


Dat $(\varphi \wedge \psi) \vee (\varphi \wedge \chi) \models \varphi \wedge (\psi \vee \chi)$ blijkt uit:

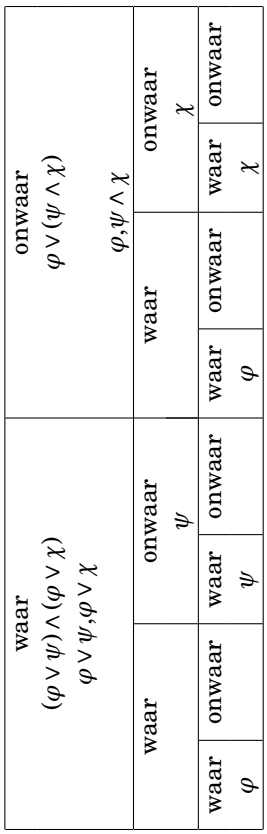


Beide tableaus sluiten: de formule-schema's zijn dus logisch equivalent. □

2. Een tableau voor $\varphi \vee (\psi \wedge \chi) \models (\varphi \vee \psi) \wedge (\varphi \vee \chi)$ is:



Een tableau voor $(\varphi \vee \psi) \wedge (\varphi \vee \chi) \models \varphi \vee (\psi \wedge \chi)$ is:



De tableaus sluiten weer: ook deze formule-schema's zijn dus logisch equivalent. □

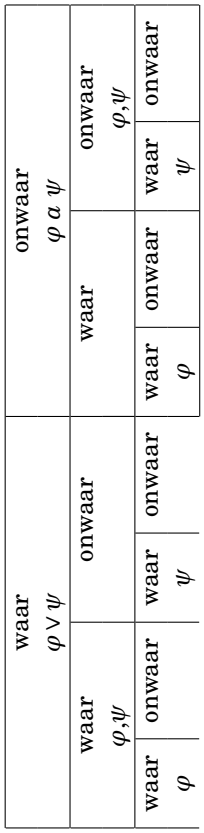
Blz. 93, it. 8.15: – α-links:

waar $\varphi\ a\ \psi$		onwaar	
waar	onwaar	waar	onwaar
φ	ψ	ψ	φ

– a-rechts:

waar		onwaar $\varphi\ a\ \psi$	
waar	onwaar	waar	onwaar
φ,ψ			ψ,φ

Blz. 94, it. 8.16:



Het linker subtableau sluit niet; de redenering is dus ongeldig.

Blz. 94, it. 8.17: – $\leftrightarrow$-links:

waar $\varphi \leftrightarrow \psi$		onwaar	
waar	onwaar	waar	onwaar
φ,ψ			φ,ψ

– $\leftrightarrow$-rechts:

waar		onwaar $\varphi \leftrightarrow \psi$	
waar	onwaar	waar	onwaar
φ	ψ	ψ	φ

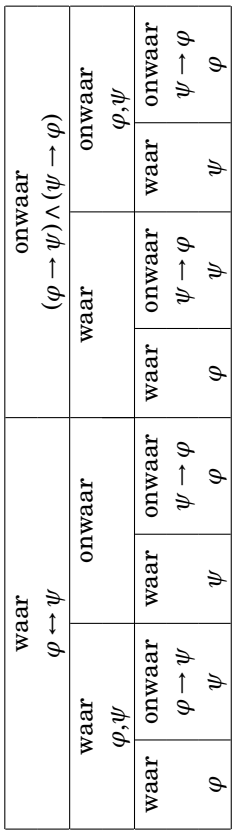
Blz. 94, it. 8.18: 1. Het bewijs van $\varphi \to (\psi \to \chi) \models (\varphi \wedge \psi) \to \chi$ luidt:

waar $\varphi \to (\psi \to \chi)$ $\varphi \wedge \psi$ φ,ψ		onwaar $(\varphi \wedge \psi) \to \chi$ χ			
waar	onwaar	waar $\psi \to \chi$		onwaar	
	φ	waar	onwaar ψ	waar χ	onwaar

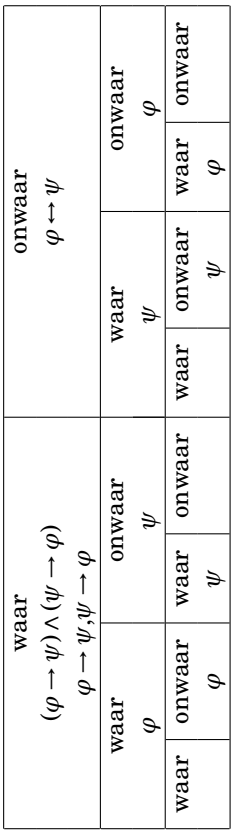
Een bewijs van $(\varphi \wedge \psi) \to \chi \models \varphi \to (\psi \to \chi)$ is:

waar $(\varphi \wedge \psi) \to \chi$ φ ψ				onwaar $\varphi \to (\psi \to \chi)$ $\psi \to \chi$ χ	
waar		onwaar $\varphi \wedge \psi$		waar χ	onwaar
waar	onwaar φ	waar	onwaar ψ		

2. Het bewijs voor de ene kant van de equivalentie is:



Voor de andere kant:



3. De ene kant van het bewijs is:

waar $\varphi \to \psi$ $\neg\psi$ φ		onwaar $\neg\psi \to \neg\varphi$ $\neg\varphi$ ψ	
waar	onwaar φ	waar ψ	onwaar

De andere kant is:

waar $\neg\psi \to \neg\varphi$ φ		onwaar $\varphi \to \psi$ ψ	
waar ψ	onwaar $\neg\psi$	waar $\neg\varphi$	onwaar φ

4. De ene kant is:

waar $\varphi \to \neg\psi$ ψ φ		onwaar $\psi \to \neg\varphi$ $\neg\varphi$	
waar	onwaar φ	waar $\neg\psi$	onwaar ψ

En de andere kant:

waar $\psi \to \neg\varphi$ φ ψ		onwaar $\varphi \to \neg\psi$ $\neg\psi$	
waar	onwaar ψ	waar $\neg\varphi$	onwaar φ

Blz. 94, it. 8.19: 1. $(p \wedge (p \to q)) \to q$ is een tautologie want:

waar $p \wedge (p \to q)$ $p, p \to q$		onwaar $(p \wedge (p \to q)) \to q$ q	
waar	onwaar p	waar q	onwaar

2. $(\neg q \wedge (p \to q)) \to \neg p$ is een tautologie want:

waar $\neg q \wedge (p \to q)$ $\neg q, p \to q$ p		onwaar $(\neg q \wedge (p \to q)) \to \neg p$ $\neg p$ q	
waar	onwaar p	waar q	onwaar

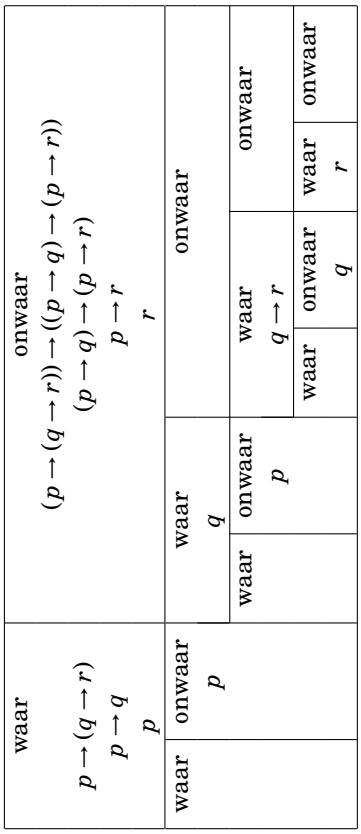
3. $((p \to q) \wedge (q \to r)) \to (p \to r)$ is een tautologie want:

waar $(p \to q) \wedge (q \to r)$ $p \to q, q \to r$ p		onwaar $((p \to q) \wedge (q \to r)) \to (p \to r)$ $p \to r$ r			
waar	onwaar p	waar q		onwaar	
		waar	onwaar q	waar r	onwaar

4. $p \to (p \to p)$ is een tautologie want:

waar	onwaar $p \to (p \to p)$
p	$p \to p$
p	p

5. $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$ is een tautologie want:



Blz. 97, it. 8.23: Op zich zijn DNFs (of CNFs) niet uniek. Bij een bestaande DNF, bijvoorbeeld

$$(p \wedge \neg q) \vee (\neg p \wedge r),$$

kunnen altijd extra literals worden ingebracht. Dat kan op het hoogste niveau, zoals in bijvoorbeeld

$$(p \wedge \neg q) \vee (\neg p \wedge r) \vee (p \wedge s \wedge \neg s),$$

maar ook op component-niveau, zoals bijvoorbeeld

$$(p \wedge \neg q \wedge s) \vee (\neg p \wedge r \wedge s) \vee (p \wedge \neg q \wedge \neg s) \vee (\neg p \wedge r \wedge \neg s).$$

(Zie hoe s afhankelijkheden introduceert.) Combinaties daarvan zijn ook mogelijk.

Sluiten we dergelijke "verdunningen" uit, dus eisen we dat onderdelen niet over de hele linie hetzelfde atoom of de ontkenning ervan mogen bevatten (eerste t.v.b.) en dat onderdelen geen complementair paar literals mogen bevatten (tweede t.v.b.), zelfs dan zijn DNFs nog niet uniek, vanwege de volgorde van de literals, i.e, $(p \wedge \neg q) \vee (\neg p \wedge r)$ is syntactisch niet gelijk aan $(\neg q \wedge p) \vee (\neg p \wedge r)$, maar wel logisch equivalent.

Eisen we ook nog dat literals in een bepaalde volgorde staan (een alfabetische volgorde ligt voor de hand), pas dan zijn DNFs (en CNFs) uniek.

Blz. 98, it. 8.24: – Dat $\{\wedge, \neg\}$ functioneel volledig is, is als volgt in te zien. Stelling 8.1 garandeert dat er bij iedere n-plaatsige operatie op $\{0,1\}$ een formule φ is die deze operatie als waarheidstabel heeft, waarbij φ hoogstens de connectieven $\neg$, $\wedge$ en $\vee$ bevat. Alleen $\vee$ willen we in het onderhavige geval er niet bij hebben; $\vee$ kunnen we echter elimineren door gebruik te maken van de volgende logische equivalentie:

$$\varphi \vee \psi \Leftrightarrow \neg(\neg\varphi \wedge \neg\psi)$$

– Voor $\{\rightarrow, \neg\}$ volgt de functionele volledigheid geheel analoog aan het vorige geval. Het enige verschil is dat we zowel $\wedge$ als $\vee$ moeten herschrijven in termen van $\rightarrow$ en $\neg$. Dit kan als volgt: (vergelijk de afkortingen uit § 9.6)

$$\begin{array}{lcl} \varphi \wedge \psi & \Leftrightarrow & \neg(\varphi \rightarrow \neg\psi) \\ \varphi \vee \psi & \Leftrightarrow & \neg\varphi \rightarrow \psi \end{array}$$

– Dat $\{\vee, \neg\}$ functioneel volledig is volgt weer analoog, maar nu met gebruik van de logische equivalentie:

$$\varphi \wedge \psi \Leftrightarrow \neg(\neg\varphi \vee \neg\psi)$$

Blz. 98, it. 8.25: We bewijzen dat $\{\wedge, \vee\}$ niet functioneel volledig is door te laten zien dat *negatie*, c.q. de functie $f : \{0,1\} \rightarrow \{0,1\}$ zodat $f(0) = 1$ en $f(1) = 0$, niet uitdrukbaar is in een formule φ die behalve één propositieletter p alleen $\wedge$ en $\vee$ mag bevatten. Met inductie naar de opbouw van zulke φ is namelijk zelfs te bewijzen dat φ dezelfde waarheidswaarde heeft als p:

– Als $\varphi = p$ dan geldt dit uiteraard.

– Stel de bewering geldt voor willekeurige formules α en β die uitsluitend met p, $\wedge$ en $\vee$ zijn opgebouwd. We moeten nu aantonen dat de bewering ook voor $\alpha \vee \beta$ en $\alpha \wedge \beta$ geldt. Dit volgt echter onmiddellijk uit de tabel:

p	α	β	$\alpha \vee \beta$	$\alpha \wedge \beta$
1	1	1	1	1
0	0	0	0	0

Blz. 98, it. 8.26: We bewijzen dat $\{\neg, \leftrightarrow\}$ niet functioneel volledig is door te laten zien dat *disjunctie, conjunctie* en *implicatie* niet uitdrukbaar zijn met alleen $\neg$ en $\leftrightarrow$, aldus:

Waarheidstafels van formules die met p, q en $\leftrightarrow$ (en evt. $\neg$) gemaakt zijn hebben altijd een *even* aantal enen (0, 2 of 4) per kolom; dit is weer met inductie te bewijzen. Disjuncties als $p \vee q$ (preciezer: de tweeplaatsige operatie die hiervan de waarheidstafel is) hebben echter een oneven aantal enen en nullen, en kunnen dus nooit uitdrukbaar zijn met behulp van $\neg$ en $\leftrightarrow$.

Blz. 98, it. 8.27: Om aan te tonen dat $\{\wedge, \rightarrow\}$ functioneel onvolledig is, moeten we zoeken naar een waarheidsfunctie die niet uitdrukbaar is met een formule waarin alleen $\wedge$ en $\rightarrow$ voorkomen. Met Stelling 8.1 volstaat het te zoeken naar een logische formule in $\{\wedge, \vee, \neg\}$ die niet kan worden uitgedrukt met $\{\wedge, \rightarrow\}$. We draaien de zaak om: om een niet-uitdrukbare formule te vinden, proberen we eerst te ontdekken welke eigenschappen $\wedge$ en $\rightarrow$ bezitten, om dan vervolgens met een formule te komen die deze eigenschap *niet* heeft.

Laten we $\wedge$ en $\rightarrow$ eens nader bekijken:

x	y	$\wedge(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$\rightarrow(x,y)$
0	0	1
0	1	1
1	0	0
1	1	1

Als we proberen een regelmatigheid te ontdekken in combinaties van $\wedge$ en $\rightarrow$, zien we al snel dat 1-tupels altijd weer op 1 afgebeeld worden. Ook met (meer ingewikkelde) combinaties van $\wedge$ en $\rightarrow$ blijft dit zo. Om dit in te zien, kunnen we een willekeurige waarheidsfunctie f louter met behulp van $\wedge$ en $\rightarrow$ construeren, en vaststellen dat het tupel $(1,1,1,1,1)$ afgebeeld wordt op 1.

$$f:\{0,1\}^5\rightarrow\{0,1\}:$$
$$(p_1,p_2,p_3,p_4,p_5)\mapsto(p_5\rightarrow p_4)\wedge((p_1\wedge p_3)\rightarrow p_2)$$

Ook bij andere waarheidsfuncties van $\{0,1\}^n$ naar $\{0,1\}$ die louter met behulp van $\wedge$ en $\rightarrow$ zijn geconstrueerd, kunnen we vaststellen dat zij $(1,\ldots,1)$ afbeeld op 1. (Eigenlijk moeten we dit netjes bewijzen,—dit doen we straks.) Dus elke waarheidsfunctie die $(1,\ldots,1)$ *niet* afbeeld op 1, kan niet uitgedrukt worden met een $\{\wedge,\rightarrow\}$-waarheidsfunctie. In het bijzonder kan de negatie niet uitgedrukt worden met een $\{\wedge,\rightarrow\}$-waarheidsfunctie, omdat de negatie een functie is van $\{0,1\}$ naar $\{0,1\}$, die 1 afbeeld op een waarde ongelijk aan 1. Informeel is het bewijs nu rond, we moeten het alleen nog netjes uitwerken:

Bewering: *een waarheidsfunctie* $f:\{0,1\}^n\rightarrow\{0,1\}$ *die louter met behulp van de functies* $\wedge$ *en* $\rightarrow$ *is geconstrueerd beeldt* $(1,\ldots,1)$ *af op* 1.

Bewijs: met inductie naar de complexiteit van f.

1. (Basis.) De meest eenvoudige $\{\wedge,\rightarrow\}$-waarheidsfuncties zijn $\wedge$ en $\rightarrow$ zelf. Uit bovenstaande waarheidstafels blijkt dat deze functies inderdaad $(1,1)$ op 1 afbeelden.
2. (Inductiestap.) Als f een samengestelde $\{\wedge,\rightarrow\}$-waarheidsfunctie is, dan is f samengesteld uit twee eenvoudiger functies g en h, hetzij als $f=g\wedge h$, hetzij als $f=g\rightarrow h$. Per inductie mogen we aannemen dat g en h elk 1-tupel afbeelden op 1. Uit bovenstaande waarheidstafels kunnen we weer aflezen dat de laatste stap, i.e., $g\wedge h$ of $g\rightarrow h$, dan ook weer in 1 resulteert.

N.B.: de meest eenvoudige $\{\wedge,\rightarrow\}$-waarheidsfuncties zijn $\wedge$ en $\rightarrow$ zelf, en niet bijvoorbeeld de projectie-waarheidsfuncties

$$P_i:\{0,1\}^n\rightarrow\{0,1\}:(x_1,\ldots,x_n)\mapsto x_i,$$

$1\leq i\leq n$, of de constante waarheidsfuncties

$$\bot:\{0,1\}^n\rightarrow\{0,1\}:(x_1,\ldots,x_n)\mapsto 0$$

en

$$T:\{0,1\}^n\rightarrow\{0,1\}:(x_1,\ldots,x_n)\mapsto 1.$$

Immers, $\{\wedge,\rightarrow,\bot\}$ is wél functioneel volledig, getuige de identiteit $\neg\varphi\equiv\varphi\rightarrow\bot$.

Blz. 98, it. 8.28: – Met de Quine dolk:

$$\begin{array}{lcl}\varphi\wedge\psi & := & (\varphi\dagger\varphi)\dagger(\psi\dagger\psi)\\ \varphi\vee\psi & := & (\varphi\dagger\psi)\dagger(\varphi\dagger\psi)\\ \varphi\rightarrow\psi & := & ((\varphi\dagger\varphi)\dagger\psi)\dagger((\varphi\dagger\varphi)\dagger\psi)\\ \varphi\leftrightarrow\psi & := & ((\varphi\dagger\varphi)\dagger\psi)\dagger((\psi\dagger\psi)\dagger\varphi)\end{array}$$

– Met de Sheffer streep:

$$\begin{array}{lcl}\varphi\wedge\psi & := & (\varphi\mid\psi)\mid(\varphi\mid\psi)\\ \varphi\vee\psi & := & (\varphi\mid\varphi)\mid(\psi\mid\psi)\\ \varphi\rightarrow\psi & := & (\varphi\mid(\psi\mid\psi))\\ \varphi\leftrightarrow\psi & := & ((\varphi\mid\varphi)\mid(\psi\mid\psi))\mid(\varphi\mid\psi)\end{array}$$

Blz. 98, it. 8.29: In principe zijn er zestien binaire operatoren. De eerste acht zijn:

x	y	$\circ_1$	$\circ_2$	$\circ_3$	$\circ_4$	$\circ_5$	$\circ_6$	$\circ_7$	$\circ_8$	…
0	0	0	0	0	0	0	0	0	0	…
0	1	0	0	0	0	1	1	1	1	…
1	0	0	0	1	1	0	0	1	1	…
1	1	0	1	0	1	0	1	0	1	…
			$\wedge$						$\vee$	…

De laatste acht zijn:

	$\circ_9$	$\circ_{10}$	$\circ_{11}$	$\circ_{12}$	$\circ_{13}$	$\circ_{14}$	$\circ_{15}$	$\circ_{16}$
…	1	1	1	1	1	1	1	1
…	0	0	0	0	1	1	1	1
…	0	0	1	1	0	0	1	1
…	0	1	0	1	0	1	0	1
…	$\dagger$	$\leftrightarrow$				$\rightarrow$	$\mid$	

(Let op de symmetrie in de verticale midden-as.)

Van $\mid$ (nand) en $\dagger$ (nor) weten we dat ze op zichzelf functioneel compleet zijn. We moeten bewijzen dat de anderen dat niet zijn.

Laten we eerst kijken welke operatoren $\circ_i$ er niet in slagen om $\neg$ uit te drukken. Dit zijn alle operatoren die gesloten zijn op $\{0\}$ of $\{1\}$. Immers deze operatoren kunnen van een 0 nooit meer een 1 maken en omgekeerd. Vanwege geslotenheid op $\{0\}$ mogen we concluderen dat $\{\circ_1,\ldots,\circ_8\}$ als verzameling operatoren incompleet is, laat staan alle deelverzamelingen, laat staan singleton-deelverzamelingen (individuele elementen). Vanwege geslotenheid op $\{1\}$ is de verzameling operatoren $\{\circ_{10},\circ_{12},\circ_{14},\circ_{16}\}$ incompleet, en daarmee ook alle deelverzamelingen. Daarmee vallen meteen twaalf operatoren af waaronder bekende binaire operatoren zoals $\wedge$, $\vee$, $\leftrightarrow$, en $\rightarrow$.

Als kandidaat-compleet blijven over $\circ_9$, $\circ_{11}$, $\circ_{13}$, en $\circ_{15}$. Twee daarvan, te weten $\circ_9$, en $\circ_{15}$ hoeven we niet verder te analyseren, want dat zijn $\dagger$ (nor) en $\mid$ (nand) waarvan we weten dat ze compleet zijn. Blijven over $\circ_{11}$ en $\circ_{13}$. Wat valt daar nog verder over op te merken? Welke regelmatigheden kunnen we vinden die bewaard (invariant) blijven onder herhaalde toepassing? Eén regelmatigheid (er zijn er meer) is dat zowel $\circ_{11}$ als $\circ_{13}$

evenveel 0-en als 1-en produceren. Omdat $\wedge$ (of $\vee$ of $\rightarrow$) deze eigenschap niet heeft, kan $\wedge$ niet worden uitgedrukt met combinaties van operatoren uit de verzameling $\{\circ_{11}, \circ_{13}\}$. Daarmee is de verzameling $\{\circ_{11}, \circ_{13}\}$ incompleet en daarmee ook alle deelverzamelingen, in het bijzonder $\{\circ_{11}\}$ en $\{\circ_{13}\}$.

Blz. 98, it. 8.30: – †-links:

waar	onwaar
$\varphi \dagger \psi$	φ, ψ

– †-rechts:

waar		onwaar $\varphi \dagger \psi$	
waar φ	onwaar	waar ψ	onwaar

– |-links:

waar $\varphi \mid \psi$		onwaar	
waar	onwaar φ	waar	onwaar ψ

– |-rechts:

waar	onwaar $\varphi \mid \psi$
φ, ψ	

Blz. 103, it. 1a:

1.	$p \wedge \neg r$	ass
2.	p	G$\wedge$, 1
3.	$\neg r$	G$\wedge$, 1
4.	$\neg r \wedge p$	I$\wedge$, 2, 3

Blz. 103, it. 1b:

1.	$p \rightarrow q$	
2.	$p \rightarrow r$	ass
3.	$\quad p$	ass
4.	$\quad q$	G$\rightarrow$, 1, 3
5.	$\quad r$	G$\rightarrow$, 2, 3
6.	$\quad q \wedge r$	I$\wedge$, 4, 5
7.	$p \rightarrow (q \wedge r)$	I$\rightarrow$, 3–6

Blz. 103, it. 1c:

1.	$p \rightarrow (p \rightarrow q)$	ass
2.	$\quad p$	ass
3.	$\quad p \rightarrow q$	G$\rightarrow$, 1, 2
4.	$\quad q$	G$\rightarrow$, 2, 3
5.	$p \rightarrow q$	I$\rightarrow$, 2–4

Blz. 103, it. 1d:

1.	$(p \wedge \neg q) \rightarrow r$	
2.	$\quad p$	ass
3.	$\quad \neg q$	ass
4.	$\quad p \wedge \neg q$	I$\wedge$, 2, 3
5.	$\quad r$	G$\rightarrow$, 1, 4
6.	$\quad r \vee s$	I$\vee$, 5
7.	$\neg q \rightarrow (r \vee s)$	I$\rightarrow$, 3–5

Blz. 103, it. 1m:

1.	$\neg p \rightarrow q$	ass
2.	$\quad \neg(p \vee q)$	ass
3.	$\quad\quad p$	ass
4.	$\quad\quad p \vee q$	I$\vee$, 3
5.	$\quad\quad \bot$	G$\neg$, 2, 4
6.	$\quad \neg p$	I$\neg$, 3–5
7.	$\quad q$	G$\rightarrow$, 1, 6
8.	$\quad p \vee q$	I$\vee$, 7
9.	$\quad \bot$	G$\neg$, 2, 8
10.	$\neg\neg(p \vee q)$	I$\neg$, 2–9
11.	$p \vee q$	$\neg\neg$-regel, 10

Blz. 103, it. 1: Stel dat $\neg\varphi \vdash \varphi$. Dan bestaat er per definitie van $\vdash$ dus een bewijs voor φ uit $\neg\varphi$:

1.	$\neg\varphi$	ass
..	$\quad \ldots$	
..	$\quad \ldots$	
n.	$\quad \varphi$	

Laten we dit even noteren met

1.	$\neg\varphi$	ass
π.	$\quad \Pi$	
n.	$\quad \varphi$	

De zin daarvan is dat we dan van de stippeltjes af zijn ($\pi = \{2, \ldots, n-1\}$ en $\Pi = \{\varphi_2, \ldots, \varphi_{n-1}\}$).

Als we nu kijken naar dit bewijs, dan zien we dat we eigenlijk een tegenspraak hebben afgeleid: we hebben φ afgeleid uit $\neg\varphi$. Laten we dit in de Fitch-afleiding zélf expliciet maken door de G$\neg$-regel toe te passen (regel $n+1$). Omdat $\bot$ is afgeleid kunnen we vervolgens concluderen dat de aanname op de 1e regel onjuist is, i.e.,

we passen de I¬-regel toe om $\neg\neg\varphi$ te concluderen en $\neg\varphi$ in te trekken. Nu zijn er geen aannames meer. De dubbele negatie mag tenslotte worden weggehaald met de ¬¬-regel.

1.	$\neg\varphi$	ass
π.	Π	
n.	φ	
$n+1$.	$\bot$	G¬, 1, n
$n+2$.	$\neg\neg\varphi$	I¬
$n+3$.	φ	¬¬-regel, $n+2$

Blz. 105, it. 9.4: 1. Kies in axioma-schema *1* $\varphi = p$ en $\psi = p$.

2. Kies in axioma-schema *1* $\varphi = p \to q$ en $\psi = p$.

3. Kies in axioma-schema *2* $\varphi = p$, $\psi = q$ en $\chi = p$.

4. Kies in axioma-schema *2* $\varphi = p$, $\psi = q$ en $\chi = p \to q$.

5. Kies in axioma-schema *3* $\varphi = \neg p$, en $\psi = q \to r$.

Blz. 106, it. 9.5:

1	$\neg\neg q \to \neg\neg p$	[premisse]
2	$(\neg\neg q \to \neg\neg p) \to (\neg p \to \neg q)$	[ax 3]
3	$\neg p \to \neg q$	[MP uit 1 en 2]
4	$(\neg p \to \neg q) \to (q \to p)$	[ax 3]
5	$q \to p$	[MP uit 3 en 4]
6	q	[premisse]
7	p	[MP uit 5 en 6]

Blz. 106, it. 9.6:

1	$p \to (q \to (r \to s))$	[premisse]
2	p	[premisse]
3	$q \to (r \to s)$	[MP uit 1 en 2]
4	q	[premisse]
5	$r \to s$	[MP uit 3 en 4]
6	r	[premisse]
7	s	[MP uit 5 en 6]

Blz. 106, it. 9.7:

1	$p \to (q \to p)$	[ax 1]
2	$(p \to (q \to p)) \to ((p \to q) \to (p \to p))$	[ax2]
3	$(p \to q) \to (p \to p)$	[MP uit 1 en 2]

Blz. 107, it. 9.8:

1	φ	[premisse]
2	$\varphi \to \psi$	[gegeven stelling]
3	ψ	[MP uit 1 en 2]

Blz. 107, it. 9.9: **Bewijs:** analoog aan het bewijs van de deductiestelling in de tekst plegen we inductie naar de lengte van de afleiding van ψ uit $\varphi_1,\ldots,\varphi_n,\varphi_{n+1}$.

- als de lengte van de afleiding 1 is dan zijn er 3 mogelijkheden:
  - ψ is een axioma dan ook $\vdash \varphi_{n+1} \to \psi$ (zie bewijs tekst) dus zeker $\varphi_1,\ldots,\varphi_n \vdash \varphi_{n+1} \to \psi$.
  - als $\psi = \varphi_{n+1}$, dan volgt het verlangde weer vanwege $\vdash \psi \to \psi$.
  - $\psi \in \{\varphi_1,\ldots,\varphi_n\}$. Dan redeneren we weer als in het eerste geval:

1	ψ	[premisse]
2	$\psi \to (\varphi_{n+1} \to \psi)$	[ax 1]
3	$\varphi_{n+1} \to \psi$	[MP uit 1 en 2]

    Dus ook in dit geval $\varphi_1,\ldots,\varphi_n \vdash \varphi_{n+1} \to \psi$.
- stel de bewering geldt voor alle formules $\varphi_1,\ldots,\varphi_{n+1},\psi$ waarvoor ψ in hoogstens k stappen uit $\{\varphi_1,\ldots\varphi_{n+1}\}$ afleidbaar is. Bekijk nu een dergelijke afleiding die precies $k+1$ stappen vergt. Omdat deze afleiding meer dan 1 stap telt, moet de laatste stap een toepassing van MP zijn, zeg uit χ en $\chi \to \psi$. De laatste twee formules zijn dus zelf in hoogstens k stappen afgeleid; dus geldt

  $$\varphi_1,\ldots,\varphi_n \vdash \varphi_{n+1} \to (\chi \to \psi) \text{ en } \varphi_1,\ldots,\varphi_n \vdash \varphi_{n+1} \to \chi$$

  Net als in het tekstbewijs volgt de gewenste conclusie hieruit met behulp van axioma 2:

  $$(\varphi_{n+1} \to (\chi \to \psi)) \to ((\varphi_{n+1} \to \chi) \to (\varphi_{n+1} \to \psi))$$

  en tweemaal Modus Ponens. □

Blz. 108, it. 9.10: – Alle axioma's zijn tautologieën. Voor axioma-schema 2 is dit reeds bewezen in opdracht 5.23.6.; voor axioma-schema 3 in 5.35.3. Voor axioma's volgens schema 1 maken we bijvoorbeeld het volgende semantische tableau:

waar	onwaar
	$\varphi \to (\psi \to \varphi)$
φ	$\psi \to \varphi$
ψ	φ

– Dat de relatie van logisch-gevolg-zijn de regel Modus Ponens respecteert, met andere woorden dat φ, $\varphi \to \psi \models \psi$ is al bewezen in opdracht 5.27, maar blijkt wellicht nog eenvoudiger uit het volgende tableau:

waar $\varphi,\varphi \to \psi$		onwaar ψ	
waar	onwaar φ	waar ψ	onwaar

– Parallel aan het deductieve bewijs (van $\Sigma \vdash \varphi$) is er dus een bewijs voor het overeenkomstige logische gevolg (van $\Sigma \models \varphi$). Dit is verder te formaliseren tot een bewijs met inductie naar de lengte van de afleiding. □

Blz. 109, it. 9.11: **Bewijs:** als Γ inconsistent is, dan is er een formule φ zodat $\Gamma \vdash \varphi$ en $\Gamma \vdash \neg\varphi$. Laat nu $\Gamma \subseteq \Gamma'$. Omdat de in de afleidingen van φ en $\neg\varphi$ gebruikte premissen uit Γ ook in Γ' voorkomen, geldt eveneens $\Gamma' \vdash \varphi$ en $\Gamma' \vdash \neg\varphi$. Dus Γ' is inconsistent. □

Blz. 109, it. 9.12: Bekijk een verzameling P van propositieletters en kies een $p \in P$, dan is $P \nvdash \neg p$ (want de waardering die alle propositieletters waar maakt toont aan dat $P \not\models \neg p$ en de correctheid van het systeem doet de rest). Dus is volgens stelling 9.7 de verzameling P consistent.

Blz. 109, it. 9.13: Laat V een valuatie zijn van de propositielogische taal $\mathscr{T}$ en noem $\Gamma = \{\varphi \in \mathscr{T} \mid V(\varphi) = 1\}$. Γ beschikt over de verlangde eigenschappen:

– Omdat V een functie is, is er geen formule φ zodat $V(\varphi) = 1$ én $V(\varphi) = 0$. Dus is er geen φ zodat $V(\varphi) = 1$ en $V(\neg\varphi) = 1$. Dus is er ook geen φ zodat $\varphi \in \Gamma$ en $\neg\varphi \in \Gamma$. Kortom, Γ is consistent.

– Stel $\Gamma \subset \Gamma'$, dan is er een $\psi \in \Gamma' - \Gamma$. Omdat $\psi \notin \Gamma$ moet $V(\psi) = 0$. Dus $V(\neg\psi) = 1$, en $\neg\psi$ behoort dus tot Γ, en derhalve ook tot Γ'. Maar $\psi \in \Gamma'$. Dus Γ' is inconsistent. □

Blz. 117, it. 10.1: 1. Sjp

2. Spj

3. Sjj

Blz. 117, it. 10.2: 1. Ish

2. $Kh \to Wp$

3. $\neg Gshp$

4. $\neg Mh$

5. $Ms \to Ish$

6. $Ish \vee Ihs$

7. $Gshs$

Blz. 119, it. 10.3: 1. Rxx

2. $\exists y \forall z Sxyz$

3. $\forall y Sxyz$

4. $\exists y Rxy$

5. $Ax \wedge Bx$

6. $Ax \to Bx$

7. $(Ax \to Bx) \wedge Cx$

8. Bx

Blz. 119, it. 10.4: – als φ een atomaire formule is die v bevat, dan is die v een vrij voorkomen van een variable (afgekort: vvv) in φ.

– als v een vvv is in φ, dan is v ook een vvv in $\neg\varphi$.

– als v een vvv is in φ of ψ, dan is v ook een vvv in $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ en $(\varphi \leftrightarrow \psi)$.

– als v een vvv is in φ en $v \neq u$, dan is v ook een vvv in $\forall u\varphi$ en $\exists u\varphi$.

– In geen ander geval is v een vvv in φ

Blz. 119, it. 10.5: De formules *6.* en *8.* zijn open; de variabele x heeft een vrij voorkomen. De andere formules zijn dus gesloten.

Blz. 120, it. 10.6: 1. $\forall x \neg \forall y \neg Rxy$

2. $\neg \forall x \neg Ax \vee \forall y By$

3. $\neg \forall x \neg \forall y \neg \forall z \neg Sxyz$

4. $\forall x \neg Ax \vee \forall y By$

5. $\neg(\neg \forall x \neg Ax \vee \forall y By)$ ofwel $\forall x \neg Ax \wedge \neg \forall y By$.

6. $\neg \forall x \forall y \forall z \neg(Ax \vee Ryz)$

Blz. 120, it. 10.7: 1. $\neg \exists x \neg \exists y Rxy$

2. $\exists x Ax \vee \neg \exists y \neg By$

3. $\exists x \neg \exists y \neg \exists z Sxyz$

4. $\neg \exists x Ax \vee \neg \exists y \neg By$

5. $\neg(\exists x Ax \vee \neg \exists y \neg By)$ ofwel $\neg \exists x Ax \wedge \exists y \neg By$

6. blijft gelijk

Blz. 122, it. 10.8: Het discussiedomein is voor *1.* t/m *4.* de verzameling mensen, en voor *5.* t/m *7.* de verzameling mensen en uitingen. Vertaalsleutel:

a :	Annie
b :	Bhagwan
j :	Jan
Vxy :	x volgt y
Zxy :	x zegt y
Axy :	x adoreert y
Px :	x is een profeet
Qx :	x is een profiteur
Txy :	x spreekt y tegen

1. $Pb \vee Qb$ (steeds: hij = Bhagwan)

2. $Pb \to Aab$

3. $\forall x(Vxb \to Axb)$

4. $\forall x(\neg Axb \to \neg Vxb)$

5. $\forall x(Zbx \to Tjx)$

6. $\forall x(Zbx \to Tjx)$

7. $\forall x(Zbx \to \neg \exists y(Vyb \wedge Tyx))$ of $\neg \exists y \exists x(Zbx \wedge Vyb \wedge Tyx)$

Blz. 122, it. 10.9: Discussiedomein: de verzameling mensen. Vertaalsleutel:

– Bx : x is barbier;

– Sxy : x scheert y.

1. $\forall x((Bx \wedge \neg Sxx) \to \exists y(By \wedge Syx))$

2. $\forall x((Bx \wedge \neg Sxx) \to \neg \exists y(By \wedge Syx))$

3. $\neg\exists x(Bx \wedge \neg Sxx \wedge \forall y(By \rightarrow Syx))$

4. $\forall x((Bx \wedge \neg Sxx) \rightarrow \neg\forall y Sxy)$

5. $\forall x((Bx \wedge \neg Sxx) \rightarrow \neg\forall y(\neg Syy \rightarrow Sxy))$

Blz. 123, it. 10.10: 1. $\forall x\forall y((Kyx \wedge Lxy) \rightarrow K'xy)$
discussiedomein: mensen
vertaalsleutel:

- Kxy : x is een kind van y;
- $K'xy$: x kastijdt y;
- Lxy : x heeft y lief.

2. $\neg\exists x(Rx \wedge \neg Dx)$
discussiedomein: planten
vertaalsleutel:

- Dx : x heeft doornen;
- Rx : x is een roos.

3. $\neg\forall x Tx$
discussiedomein: hout
vertaalsleutel: Tx : x is timmerhout.

Andere mogelijkheid:
$\neg\forall x(Hx \rightarrow Tx)$
discussiedomein: voorwerpen
vertaalsleutel:

- Hx : x is van hout;
- Tx : x is van timmerhout.

4. $\neg\forall x(Bx \rightarrow Gx)$
discussiedomein: voorwerpen
vertaalsleutel:

- Bx : x blinkt;
- Gx : x is van goud.

5. $\forall x\forall y\forall z((Kx \wedge Vxyz) \rightarrow Ozx)$
discussiedomein: mensen en uitingen
vertaalsleutel:

- Kx : x is een kind;
- Oxy : x slaat y over;
- $Vxyz$: x vraagt y aan z.

6. $\neg\forall x(\exists y(Ly \wedge My \wedge Dxy) \rightarrow Kx)$
discussiedomein: mensen en messen; vertaalsleutel:

Dxy :	x draagt y	Kx :	x is kok
Lx :	x is lang	Mx :	x is een mes

Blz. 123, it. 10.11: 1. $\forall x(Ax \rightarrow Bx)$

2. $\forall x(Ax \leftrightarrow Bx)$

3. $(\forall x(Ax \rightarrow Bx) \wedge \forall x(Bx \rightarrow Cx)) \rightarrow \forall x(Ax \rightarrow Cx)$

Blz. 123, it. 10.12: 1. $\neg\exists x Oxx$

2. $\forall x\exists y(My \wedge Oyx)$

3. $\forall x\exists y(\neg My \wedge Oyx)$

4. $\exists x\exists y Oxy$

5. $\neg\forall x\exists y(Mx \wedge Oxy)$

6. $\neg\forall x\exists y\exists z(\neg My \wedge Mz \wedge Oxy \wedge Oxz)$

7. $\forall x\exists y\exists z(Oyz \wedge Ozx \wedge \neg My)$

8. $\neg\forall x\exists y\exists z(Oxz \wedge Ozy)$

Blz. 123, it. 10.13: 1. 'Iemand is een kind van Marie.' of: 'Marie heeft een kind.'

2. 'Piet is een kleinkind van Marie.'

3. 'Marie is Piet's grootmoeder van vaderskant.'

4. 'Marie en Piet hebben samen een dochter.'

5. 'Marie heeft een kind en van al haar kinderen is Piet de vader.'

6. 'Marie heeft een zoon, maar deze heeft geen vader.'

Blz. 123, it. 10.14: 1.

$$\frac{\begin{array}{l}\forall x(Mx \rightarrow Sx)\\ Ms\end{array}}{Ss}$$

discussiedomein: sterfelijke en onsterfelijke wezens
vertaalsleutel:

- Mx : x is een mens;
- Sx : x is sterfelijk;
- s : Sokrates.

2.

$$\frac{\begin{array}{l}\neg\exists x(Vx \wedge Zx)\\ \forall x(Kx \rightarrow Zx)\end{array}}{\neg\exists x(Kx \wedge Vx)}$$

discussiedomein: dieren
vertaalsleutel:

- Kx : x is een knaagdier;
- Vx : x is een vis;
- Zx : x is een zoogdier.

Blz. 125, it. 11.1: 1. Onwaar, want 1 staat niet in relatie tot zichzelf.

2. Waar, want 2 staat in relatie tot zichzelf.

3. Waar, want vanuit elk object loopt een pijl naar een object.

4. Onwaar, want er is geen enkel object waaruit pijlen lopen naar alle objecten.

5. Onwaar, want uit 1 loopt wel een pijl naar 2, maar niet naar zichzelf.

6. Waar, want 1 staat niet in de relatie tot zichzelf.

7. Onwaar, vergelijk *5*.

8. Waar, want voor 2 staat in relatie tot zichzelf, en naar 2 loopt een pijl.

Blz. 125, it. 11.2:

1. onwaar	4. onwaar	7. onwaar
2. onwaar	5. onwaar	8. waar.
3. waar	6. waar	

Blz. 126, it. 11.3:

1. waar	5. onwaar	9. onwaar
2. onwaar	6. waar	10. waar
3. waar	7. onwaar	11. waar
4. waar	8. onwaar	12. waar

Blz. 126, it. 11.4:

1. waar	5. waar	9. waar
2. onwaar	6. onwaar	10. waar
3. onwaar	7. onwaar	11. onwaar
4. onwaar	8. waar	12. waar

Blz. 126, it. 11.5: – $I(a) = 1$; $I(b) = 2$; $I(c) = 3$;

– $I(P) = \{3\}$

– $I(R) = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,2\rangle\}$

Blz. 126, it. 11.6:

1. waar	4. waar	7. onwaar	10. onwaar
2. onwaar	5. waar	8. waar	11. onwaar
3. waar	6. onwaar	9. onwaar	12. waar

Blz. 127, it. 1: Er bestaat een lege verzameling ($\emptyset$).

Blz. 127, it. 2: Geen verzameling heeft zichzelf als element (met de terminologie van Hoofdstuk 4: elke verzameling heeft de Russell-eigenschap).

Blz. 127, it. 3: $V \in W \Rightarrow W \notin V$ (dus: '$\in$' is asymmetrisch).

Blz. 127, it. 4: Elke niet-lege verzameling is zelf element van een verzameling.

Blz. 128, it. 11.8: $\mathscr{M}, b \models \emptyset$ als alle formules in Γ vervult worden door $\mathscr{M}, b$. Deze voorwaarde wordt ledigerwijs vervult.

Blz. 129, it. 1: Stel φ is een contradictie. Dan is φ onwaar met betrekking tot alle modellen $\mathscr{M}$. Dus $M, b \not\models \varphi$ voor alle modellen $\mathscr{M}$ en voor alle bedelingen b. Maar dan is φ niet vervulbaar. Omgekeerd gaat de redenering ook op:

φ onvervulbaar
$\Rightarrow$ $M, b \not\models \varphi$ voor alle $\mathscr{M}$ en voor alle b
$\Rightarrow$ φ is onwaar met betrekking tot alle modellen
$\Rightarrow$ φ is een contradictie.

I.p.v. "$\Rightarrow$" hadden we dus net zo goed overal gelijk "$\Leftrightarrow$" kunnen zetten. Wees hier wel voorzichtig mee.

Blz. 129, it. 2: Volgt direct uit het voorgaande item.

Blz. 129, it. 3: Stel φ is contingent. Dan is deze waar noch onwaar. Het niet onwaar zijn impliceert vervulbaarheid.

Blz. 129, it. 4: Dit spreekt dachten wij voor zich.

Blz. 129, it. 5: Als φ contradictoir is, dan is er een model, $\mathscr{M}$, zo dat $\mathscr{M}, b \not\models \varphi$ voor alle b. Dus zeker niet $\mathscr{M}, b \models \varphi$ voor alle b. Dus φ is niet waar.

Blz. 129, it. 6: Elke contingente formule tegelijkertijd niet waar en niet onwaar. Dus elke contingente formule is een tegenvoorbeeld voor deze implicatie.

Concreet tegenvoorbeeld: $\forall x Rxx$ is niet (altijd) waar: neem maar een model, $\mathscr{M}$, waar R als niet-reflexief wordt geïnterpreteerd. Maar $\forall x Rxx$ is niet onwaar: neem maar een model waar R als reflexief wordt geïnterpreteerd.

Blz. 129, it. 11.10: Per definitie:

– φ is niet waar $\Leftrightarrow$ er is een model $\mathscr{M}$ en een b, zó dat $\mathscr{M}, b \not\models \varphi$.

– φ is niet onwaar $\Leftrightarrow$ er is een model $\mathscr{M}$ is er een b, zó dat $\mathscr{M}, b \models \varphi$.

– φ is contingent $\Leftrightarrow$ φ is waar noch onwaar.

Dus:

φ is contingent
$\Leftrightarrow$ φ is niet onwaar en φ is niet waar
$\Leftrightarrow$ er is een model $\mathscr{M}_1$ en een b_1, zó dat $\mathscr{M}_1, b_1 \models \varphi$,
en er is een model $\mathscr{M}_2$ en een b_2, zó dat $\mathscr{M}_2, b_2 \not\models \varphi$.

Blz. 129, it. 11.11: Twee formules $\varphi, \psi \in \mathrm{SEN}(L)$ heten *logisch equivalent* als

voor alle modellen $\mathscr{M}$: $\mathscr{M} \models \varphi \quad \Leftrightarrow \quad \mathscr{M} \models \psi$

Blz. 129, it. 11.12:

1. $\exists x Rxx \wedge Pa$	5. $\exists x \exists y Rxy \wedge Pb$
2. $\exists x Rxb \wedge Px$	6. $\forall x \forall y Rxy \rightarrow Px$
3. $\exists x \exists y Rxy \rightarrow Px$	7. $\forall x \forall y Rxy \rightarrow Pb$
4. $\exists x \exists y (Rxy \wedge Py)$	8. $\exists x Px \wedge \exists y Ray$

Blz. 129, it. 1:

– Voor een constante c: $[e/v]c = c$.

– Voor een variabele u:

$$[e/v]u = \begin{cases} e & \text{als } u = v, \\ u & \text{als } u \neq v. \end{cases}$$

– Voor een functieterm wordt substitutie inductief toegepast:

$$[e/v]f(x_1, \ldots, x_n) = f([e/v]x_1, \ldots, [e/v]x_n).$$

Blz. 129, it. 2:

- $[e/v]R(t_1,\ldots,t_n) = R([e/v]t_1,\ldots,[e/v]t_n)$
- $[e/v]\neg\varphi = \neg[e/v]\varphi$
- $[e/v](\varphi \to \psi) = ([e/v]\varphi \to [e/v]\psi)$ en analoog voor $\wedge$, $\vee$, $\leftrightarrow$.
- $[e/v]\forall u\varphi = \begin{cases} \forall u\varphi & \text{als } u = v, \\ \forall u[e/v]\varphi & \text{als } u \neq v. \end{cases}$

Blz. 130, it. 11.14: $Hpa \wedge \exists x(Hax \wedge \neg x = p)$

Blz. 130, it. 11.15: De formule is onwaar. Immers, behalve Piet houdt ook individu 3 alleen van Annie.

Blz. 130, it. 11.16: Nu is de formule wel waar. Annie houdt behalve van zichzelf ook nog van iemand anders, en individu 3 houdt alleen van zichzelf. Alleen Piet houdt alleen van Annie.

Blz. 130, it. 11.17: 1. waar; 'Iedereen houdt van iemand' (niet noodzakelijk voor iedereen dezelfde persoon).

2. onwaar; 'Iedereen houdt van iemand anders dan zichzelf.'
3. waar; 'Wie niet van zichzelf houdt is zielig.'
4. waar; 'Iemand is zielig en houdt van een ander.'
5. waar; 'Iedereen die van zichzelf houdt, houdt van geen ander.'
6. waar; 'Iedereen die van iemand anders houdt is zielig.'

Blz. 131, it. 11.18: De lezing waarin de negatie klein bereik heeft is alleen waar in het model linksonder; de andere lezing is waar in alle modellen behalve die rechtsboven.

Blz. 131, it. 1:

$$\forall xyz((N(x) \wedge A(y) \wedge A(z) \wedge H(x,y) \wedge H(x,z)) \to y = z)$$

Blz. 131, it. 2: Persoon x bezit minstens één fiets:

$$\exists y(F(y) \wedge H(x,y))$$

Persoon x bezit hoogstens twee auto's:

$$\forall uvw((A(u) \wedge A(v) \wedge A(w) \wedge H(x,u) \wedge H(x,v) \wedge H(x,w)) \to (u = v \vee u = w \vee v = w))$$

Iedere Nederlander bezit minstens één fiets en hoogstens twee auto's.

$$\forall x(N(x) \to [\exists y(F(y) \wedge H(x,y)) \wedge \forall uvw((A(u) \wedge A(v) \wedge A(w) \wedge H(x,u) \wedge H(x,v) \wedge H(x,w)) \to [u = v \vee u = w \vee v = w])])$$

Blz. 132, it. 11.20: 1. Object 4

2. Object 4
3. Object 3
4. Object 2

Blz. 133, it. 11.21: 1. Waar; neem bijvoorbeeld $x = 2$.

2. Waar; neem bijvoorbeeld $x = 3$.
3. Onwaar; de disjunctie geldt niet als $x = 2$.
4. Waar; de conjunctie is waar als $x = 3$.
5. Onwaar; voor $x = 3$ geldt de implicatie immers niet.
6. Waar; de objecten waarvoor P geldt zijn 1 en 3; voor $x = 1$ geldt Pfx niet dus klaar, voor $x = 3$ geldt Pfx wel, maar $Pffx$ ook.

Blz. 133, it. 2: O1, O2, O3, O4, O5, EN O9.

Blz. 134, it. 1: (P1) $\forall x x + 1 \neq 0$);

(P2) $\forall xy[x + 1 = y + 1 \to x = y]$;

(P3) $\forall x(x + 0 = x)$, en $\forall xy[x + (y + 1) = (x + y) + 1]$;

(P4) $\forall x(x \cdot 0 = 0)$, en $\forall xy[x \cdot (x + 1) = (x \cdot y) + x]$;

(P5) voor elke formule φ de universele afsluiting van

$$(\varphi(0) \wedge \forall x[\varphi(x) \to \varphi(x + 1)]) \to \forall y\varphi(y).$$

Blz. 134, it. 2: Inductiebasis: $0 + 1 = 0$. Dit is een instantie van axioma P1. Inductiestap:

$$x + 1 \neq x \Rightarrow (x + 1) + 1 \neq (x + 1).$$

De contrapositie hiervan, $(x + 1) + 1 = (x + 1) \Rightarrow x + 1 = x$, is een instantie van P2.

Blz. 134, it. 3: Inductiebasis: we bewijzen de basis voor het getal $s(0)$ oftewel $0 + 1$. (Verder vereenvoudigen kan niet onmiddellijk in de Peano rekenkunde.) Te bewijzen:

$$0 + 1 \neq 0 \to \exists y(0 + 1 = y + 1).$$

De consequent is onvoorwaardelijk (zonder ons druk te maken over de waarheid van de antecedent) waar: neem $y = 0$. Inductiestap: Stel

$$x \neq 0 \to \exists y(x = y + 1).$$

Te bewijzen

$$x + 1 \neq 0 \to \exists y(x + 1 = y + 1).$$

Weer is de consequent onvoorwaardelijk waar: neem $x = y$.

Blz. 135, it. 1: Voor elk getal is er een kleiner ander getal: $\forall x \exists y(y < x)$. Immers, het getal 1 bezit geen voorganger.

Blz. 135, it. 2: Waar in $Q^<$ maar niet in $Z^<$: voor elk tweetal getallen is er een ander getal dat er tussen ligt:

$$\forall xy(x < y \rightarrow \exists z(x < z \wedge z < y)). \tag{14}$$

De ontkenning is dus waar in $Z^<$ maar niet in $Q^<$:

$$\neg\forall xy(x < y \rightarrow \exists z(x < z \wedge z < y)).$$

Blz. 135, it. 3: In de vorige onderdelen vond we een zin, φ, die waar is in $Z^<$ maar niet in $N^<$, en een zin, ψ, die waar is in $Z^<$ maar niet in $Q^<$. De zinnen $\neg\varphi$ en $\neg\psi$ zijn dus respectievelijk waar in $N^<$ maar niet in $Z^<$, en waar in $Q^<$ maar niet in $Z^<$. De conjunctie $\neg\varphi \wedge \neg\psi$ is dus waar in $N^<$ en $Q^<$ maar niet waar in $Z^<$. Desgewenst kan deze zin worden vereenvoudigd.

Blz. 135, it. 4: We gaan eerst een formule construeren die niet waar is in $Q^<$. Van $\mathbb{Q}$ is bekend dat veel verzamelingen geen kleinste bovengrens bezitten. De verzameling

$$d = \{x \in \mathbb{Q} \mid x < \sqrt{2}\},$$

bijvoorbeeld, bezit geen kleinste bovengrens: elke breuk onder de $\sqrt{2}$ is geen bovengrens, en elke breuk boven de $\sqrt{2}$ is een bovengrens, maar geen kleinste bovengrens. Laat $D/1$ een unaire predikaatletter zijn met extensie $(D)_I = d$. Dus

$$\mathscr{M}, b \models D(x) \;\Leftrightarrow\; b(x) \in d.$$

Om deze reden is de volgende formule onwaar in $Q^<$:

$$\exists z\forall x(x < z \leftrightarrow D(x)).$$

Er was net immers uitgelegd dat zo'n D-definiërende bovengrens z niet kan bestaan. Noem deze formule χ. De formule χ is wel waar in $N^<$ en $Q^<$ (want neem $z = 2$), dus we moeten nog formules vinden die niet waar zijn in $N^<$ en $Z^<$. Dat is (14). Laten we deze formule ω noemen. Dus $\chi \wedge \omega$ is niet waar in zowel $N^<$, $Q^<$, als $Z^<$.

Tenslotte hopen we aan te tonen dat $\chi \wedge \omega$ geen contradictie is. Hiervoor is het voldoende te tonen dat deze formule vervulbaar is. Welnu, deze formule is natuurlijk vervulbaar in $R^< = \langle \mathbb{R}, < \rangle$. Voor z in χ kan dan $\sqrt{2}$ genomen worden, en ω is natuurlijk ook waar in $R^<$.

Blz. 143, it. 1:

(1) $\forall xPxb \circ \forall xyPxy$
|
rechts-$\forall$ instantiatie van (1)
(2) $\circ \forall yPay$
|
rechts-$\forall$ instantiatie van (2)
$\circ Pac$
|
links-$\forall$ instantiatie van (1)
$Pab \circ$
|
links-$\forall$ instantiatie van (1)
$Pbb \circ$
|
links-$\forall$ instantiatie van (1)
$Pcb \circ$
tableau blijft open.

Dus universele formules links en existentiële formules rechts kunnen meerdere keren geïnstantieerd worden met willekeurige termen. Universele formules rechts en existentiële formules links kunnen hoeven slechts één keer geïnstantieerd te worden, en dat moet met een constante die nog niet eerder voorkwam in die tak.

Tegenmodel: $D = \{a, b, c\}$, interpreteer a, b en c als zichzelf en

$$P = \{(a, b), (b, b), (c, b)\}.$$

Dit geeft een interpretatie I. Gemakkelijk is na te gaan dat $I \models \forall xPxb$ maar $I \not\models \forall xyPxy$.

Blz. 143, it. 2:

(1) $\forall xyPxy \circ \forall xPxb$
|
rechts-$\forall$ instantiatie van (1)
(2) $\circ Pab$
|
links-$\forall$ instantiatie van (1)
(3) $\forall yPay \circ$
|
links-$\forall$ instantiatie van (3)
(4) $Pab \circ$
tableau sluit
op (2) en (4).

Blz. 143, it. 3:

(1) $\exists xyPxy \circ \exists xPxb$
|
links-$\exists$ instantiatie van (1)
(2) $\exists yPay \circ$
|
links-$\exists$ instantiatie van (2)
(3) $Pac \circ$
|
rechts-$\exists$ instantiatie van (1)
(4) $\circ Pab$
|
rechts-$\exists$ instantiatie van (1)
(5) $\circ Pbb$
|
rechts-$\exists$ instantiatie van (1)
(4) $\circ Pcb$
het tableau blijft open.

Tegenmodel: $D = \{a, b, c\}$, interpreteer a, b en c als zichzelf en

$$P = \{(a, c)\}.$$

Blz. 143, it. 4:

(1) $\exists xPxc \circ \exists xPxb$
|
links-$\exists$ instantiatie van (1)
(2) $Pac \circ$
|
rechts-$\exists$ instantiatie van (1)
(3) $\circ Pab$
|
rechts-$\exists$ instantiatie van (1)
(4) $\circ Pbb$
|
rechts-$\exists$ instantiatie van (1)
(5) $\circ Pcb$
tableau blijft open.

Tegenmodel: $D = \{a, b, c\}$, interpreteer a, b en c als zichzelf en

$$P = \{(a, c)\}.$$

Dit geeft een interpretatie I. Gemakkelijk is na te gaan dat $I \models \exists x Pxc$ maar $I \not\models \exists x Pxb$.

Blz. 143, it. 12.4: 1.

	waar		onwaar	
	$\forall x(Ax \to Bx), \exists x Ax$		$\exists x Bx$	
$\exists$-links:	Ad_1			
$\exists$-rechts:			Bd_1	
$\forall$-links:	$Ad_1 \to Bd_1$			
	waar	onwaar	waar	onwaar
$\to$-links:		Ad_1	Bd_1	

Het tableau sluit; de redenering is geldig.

2.

	waar		onwaar		
			$\forall x(((Ax \to Bx) \wedge (Bx \to Cx)) \to (Ax \to Cx))$		
$\forall$r:			$((Ad_1 \to Bd_1) \wedge (Bd_1 \to Cd_1)) \to (Ad_1 \to Cd_1)$		
$\to$r:	$(Ad_1 \to Bd_1) \wedge$				
	$(Bd_1 \to Cd_1)$		$Ad_1 \to Cd_1$		
$\wedge$l:	$Ad_1 \to Bd_1$,				
	$Bd_1 \to Cd_1$				
$\to$r:	Ad_1		Cd_1		
	waar	onwaar	waar	onwaar	
$\to$l:		Ad_1	Bd_1		
			waar	onw	waar	onwaar
$\to$l:				Bd_1	Cd_1	

Het tableau sluit; de redenering is dus geldig.

3.

	waar		onwaar	
	$\forall x(Ax \to Bx), \neg\exists x Bx$		$\neg\exists x Ax$	
$\neg$-rechts:	$\exists x Ax$			
$\exists$-links:	Ad_1			
$\neg$-links:			$\exists x Bx$	
$\exists$-rechts:			Bd_1	
$\forall$-links:	$Ad_1 \to Bd_1$			
	waar	onwaar	waar	onwaar
$\to$-links:		Ad_1	Bd_1	

Het tableau sluit; de redenering is dus geldig.

4.

	waar		onwaar	
	$\forall x(Ax \to Bx), \neg\exists x Ax$		$\neg\exists x Bx$	
$\neg$-rechts:	$\exists x Bx$			
$\exists$-links:	Bd_1			
$\neg$-links:			$\exists x Ax$	
$\exists$-rechts:			Ad_1	
$\forall$-links:	$Ad_1 \to Bd_1$			
	waar	onwaar	waar	onwaar
$\to$-links:		Ad_1	Bd_1	

Het tableau sluit niet. De twee open takken leveren hetzelfde tegenvoorbeeld: $M = \langle D, I \rangle$ met $D = \{1\}$, $I(A) = \emptyset$ en $I(B) = 1$.

5.

(1) $\exists x \neg Ax \circ \forall x(\forall y(Rxy \to Ax) \to \forall y Rxy)$

links-$\exists$ instantiatie van (1)

(2) $\neg Aa \circ$

links-$\neg$ regel (2)

(3) $\circ\, Aa$

rechts-$\forall$ instantiatie van (1)

(4) $\circ\, \forall y(Rby \to Ab) \to \forall y Rby$

rechts-$\to$ regel (4)

(5) $\forall y(Rby \to Ab) \circ \forall y Rby$

rechts-$\forall$ instantiatie van (5)

(6) $\circ\, Rbc$

links-$\forall$ instantiatie van (5)

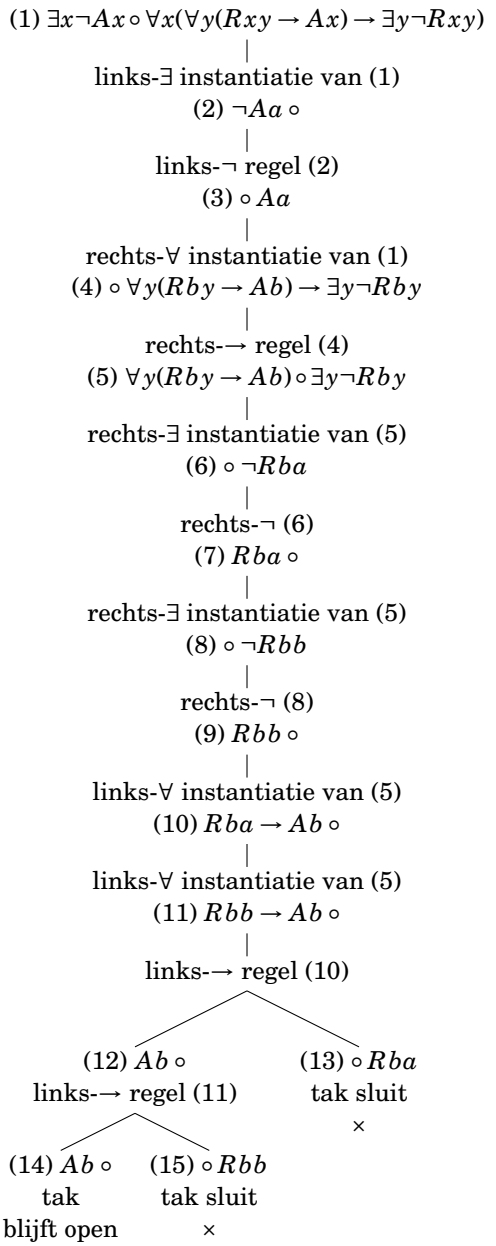
(7) $Rbc \to Ab \circ$

links-$\to$ regel (7)

$Ab \circ$	$\circ\, Rbc$
tableau blijft open	tableau blijft open

De rechtertak geeft een tegenmodel met $D = \{a, b, c\}$. Interpreteer constanten als zichzelf, en neem alle predikaten en relaties leeg.

6.



De rechtertak geeft een tegenmodel met $D = \{a, b\}$. Interpreteer $A = \{b\}$ en $R = \{(b,a),(b,b)\}$. Dit geeft een interpretatie I. Makkelijk is in te zien dat $I \models \exists x\neg Ax$, immers, daar zorgt b voor.

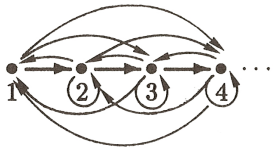
Ook is in te zien dat

$$I \not\models \forall x(\forall y(Rxy \to Ax) \to \exists y\neg Rxy).$$

Immers neem $x = b$. Dan geldt $I \models \forall y(Rby \to Ax)$ want de consequent van deze implicatie is altijd waar. En ook geldt $I \not\models \exists y\neg Rxy$. Immers, de projectie van R's eerste coördinaat is juist wel gelijk aan heel D.

De uitleg in de vorige alinea betrof slechts een contrôle. Eigenlijk is de opengebleven tak zélf het beste argument voor het bestaan van een tegenmodel.

Blz. 143, it. 12.11: Alleen bij punt 1 is er geen pijl die naar 1 zelf gaat; dit is het enige dat in het tableau verboden is, de rest maakt niet uit. Het begin van het tegenmodel ziet er dus zo uit:



Blz. 144, it. 12.13: Voor de behandeling van individuele constanten is de volgende regel nodig:

> *constanten-substitutie*: voer voor elke constante c een nieuwe d_i in, en vervang overal in het tableau c door d_i.

De redenering ziet er predikaatlogisch als volgt uit:

$$\neg\exists x(Gx \wedge Vxx), Gs \models \neg Vss$$

waarin

- Gx: x is een Griek;
- Vxy: x veracht y;
- s: Sokrates.

We testen de geldigheid van deze redenering:

	waar		onwaar	
	$\neg\exists x(Gx \wedge Vxx), Gs$		$\neg Vss$	
constanten-substitutie:	Gd_1		$\neg Vd_1d_1$	
$\neg$-rechts:	Vd_1d_1			
$\neg$-links:			$\exists x(Gx \wedge Vxx)$	
$\exists$-rechts:			$Gd_1 \wedge Vd_1d_1$	
	waar	onwaar	waar	onwaar
$\wedge$-rechts:		Gd_1		Vd_1d_1

Het tableau sluit; de redenering is geldig.

Blz. 144, it. 12.14: Voor de geldigheid van een redenering maakt het niet uit of je naar een zeker individu verwijst met een *eigennaam* (de constante) of met een *pronomen* (de variabele). De regel is dus: behandel vrije variabelen als constanten.

Blz. 146, it. 1:

$\forall x(fffx = x), \forall z(ffz = fz) \circ_1 \forall y(fy = y)$
rechts-$\forall$, 1, $[d/y]$
$\circ_2 fd = d$
links-$\forall$, 1:1, $[d/x]$
$fffd = d \circ_3$
links-$\forall$, 1:2, $[d/z]$
$ffd = fd \circ_4$
links-$\forall$, 1:2, $[f(d)/z]$
$fffd = ffd \circ_5$
pas 5 toe in 3
$ffd = d \circ_6$
pas 4 toe in 6
$fd = d \circ$
×

Blz. 146, it. 1:

$$\forall xy(fx = gy) \circ_1 \forall xy(fx = fy)$$
rechts-$\forall$, $[d/x, e/y]$
$$\circ_2 fd = fe$$
links-$\forall$, 1, $[d/x, d/y]$
$$fd = gd \circ_3$$
links-$\forall$, 1, $[e/x, d/y]$
$$fe = gd \circ_4$$
=-id
$$fe = fe \circ_5$$
pas 4 toe in 5
$$gd = fe \circ_6$$
pas 6 toe in 3
$$fd = fe \circ$$
$\times$

Blz. 146, it. 1: Deze werd voorgedaan in een eerder voorbeeld.

Blz. 153, it. 1a: Eenduidig: elke variabele speelt z'n eigen rol. Dat wil zeggen: elke kwantor heeft z'n eigen variabele, en vrije variabelen komen nooit als kwantor-variabele voor.

Blz. 153, it. 1b: De variabele x speelt twee rollen: één keer in $\forall xPx$ (scope Px) en één keer in $\forall xQx$. Niet eenduidig.

Blz. 153, it. 1c: Niet eenduidig: de variabele x speelt twee rollen: één keer als vrije variabele, en één als gekwantificeerde variabele. (Hoewel de gekwantificeerde variabele verder niet meer in de formule zelf voorkomt.)

Blz. 153, it. 1d: Niet eenduidig: de variabele x speelt twee rollen: één keer als vrije variabele in $\exists u\exists v(Rux \wedge Rxv)$, en één als gekwantificeerde variabele in $\forall x(Rxy \vee Ryx)$.

Blz. 153, it. 1e: Niet eenduidig: de variabele x wordt opnieuw gekwantificeerd in de scope van x.

Blz. 153, it. 1f: Eenduidig, want de vrije variabele x speelt één rol. De betekenis van de formule zou anders worden als tenminste één voorkomen van x zou worden vervangen door een andere variabele.

Blz. 153, it. 2a: Er wordt beweerd: een formule zonder variabelen is eenduidig. Dit is waar. Intuïtief is dit meteen in te zien: een formule is eenduidig als en slechts elke variabele in die formule z'n eigen rol speelt. Omdat er geen variabelen zijn, is die voorwaarde leeg vervuld.

Formeel bewijs: laat φ willekeurig zodanig dat $\mathrm{V}(\varphi) = \emptyset$. Dan is meteen aan de eerste voorwaarde van eenduidigheid voldaan, te weten conditie a) van Definitie 13.6.1.

We moeten ook nog controleren dat aan *precies één* voorwaarde is voldaan. Wel, dat zit snor. Conditie b) geldt niet omdat $\mathrm{VV}(\varphi)$ een deelverzameling is van V(phi) en dus ook leeg is, en conditie c) geldt niet omdat als φ een x-kwantor had, dan $x \in \mathrm{V}(\varphi)$ zou gelden waaruit zou volgen $\mathrm{V}(\varphi) \neq \emptyset$.

Blz. 153, it. 2b: Er wordt beweerd: een formule zonder vrije variabelen is eenduidig. Dit is onwaar. Kijk maar naar $\forall xPx \wedge \forall xQx$. In deze formule zijn alle variabelen gebonden maar het is duidelijk dat variabele x twee rollen speelt: één keer in $\forall xPx$ en één keer in $\forall xQx$.

Formeel gesproken voldoet $\forall xPx \wedge \forall xQx$ niet aan conditie c) van Definitie 13.6.1.

Blz. 153, it. 13.7:

1. $\forall x(Rxy \to Ryx)$ is al eenduidig.
2. $\forall xPx \wedge \forall xQx$ wordt bijvoorbeeld $\forall xPx \wedge \forall yQy$.
3. $Px \wedge \forall xy(Ty \vee \exists uPu)$ wordt bijvoorbeeld $Px \wedge \forall zy(Ty \vee \exists uPu)$.
4. $\forall x(Rxy \vee Ryx) \wedge \exists u\exists v(Rux \wedge Rxv)$ wordt bijvoorbeeld $\forall x(Rxy \vee Ryx) \wedge \exists u\exists v(Ruz \wedge Rzv)$.
5. $\forall x(\exists yRxy \to \exists x\exists zRxz)$ wordt bijvoorbeeld $\forall x(\exists yRxy \to \exists u\exists zRuz)$.
6. $Px \wedge \exists uRxu \wedge \neg Qx$ is al eenduidig.

Blz. 155, it. 13.9:

1. $\forall x(Rxy \to Ryx)$ is al eenduidig. De kwantoren staan al aan de voorkant, maar de matrix (binnen-formule) staat nog niet in CNF. Gebruikmakend van de propositie-identiteit $P \to Q \equiv \neg P \vee Q$ komen we uit op de prefix-normaalvorm $\forall x(\neg Rxy \vee Ryx)$.
2. $\forall xPx \wedge \forall xQx$ eenduidig maken levert $\forall xPx \wedge \forall yQy$. Kwantoren naar voren halen levert $\forall xy(Px \wedge Qy)$. De matrix staat al in CNF.
3. $Px \wedge \forall xy(Ty \vee \exists uPu)$ eenduidig maken levert $Px \wedge \forall zy(Ty \vee \exists uPu)$. Kwantoren naar voren halen levert $\forall zy\exists u(Px \wedge (Ty \vee Pu))$. De matrix staat al in CNF.
4. $\forall x(Rxy \vee Ryx) \wedge \exists u\exists v(Rux \wedge Rxv)$ eenduidig maken levert
$$\forall x(Rxy \vee Ryx) \wedge \exists u\exists v(Ruz \wedge Rzv).$$
Kwantoren naar voren halen levert
$$\forall x\exists u\exists v((Rxy \vee Ryx) \wedge (Ruz \wedge Rzv))$$
oftewel
$$\forall x\exists u\exists v((Rxy \vee Ryx) \wedge Ruz \wedge Rzv).$$
De matrix staat al in CNF.
5. $\forall x(\exists yRxy \to \exists x\exists zRxz)$ eenduidig maken levert $\forall x(\exists yRxy \to \exists u\exists zRuz)$. Kwantoren naar voren:
$$\forall x(\exists yRxy \to \exists u\exists zRuz)$$
$$\forall x\forall y(Rxy \to \exists u\exists zRuz)$$
$$\forall x\forall y\exists u(Rxy \to \exists zRuz)$$
$$\forall x\forall y\exists u\exists z(Rxy \to Ruz).$$

Let er op dat kwantoren die uit de antecedent van een implicatie naar voren worden gehaald, omklappen (van universeel naar existentieel, en omgekeerd). In CNF zetten:

$$\forall xy\exists uz(\neg Rxy \vee Ruz).$$

6. $Px \wedge \exists uRxu \wedge \neg Qx$ is al eenduidig. Kwantoren naar voren: $\exists u(Px \wedge Rxu \wedge \neg Qx)$. De matrix staat al in CNF.

Blz. 156, it. 13.10: Stelling: *Voor elke formule φ en fris functiesymbool f geldt dat $\forall x_1,\ldots,x_n\exists y\varphi$ vervulbaarheids-equivalent is met $\forall x_1,\ldots,x_n[f(x_1,\ldots,x_n)/y]\varphi$*

Bewijs: Voor het gemak noteren we $x_1,\ldots,x_n$ als $\bar{x}$ en $d_1,\ldots,d_n$ als $\bar{d}$. In dit bewijs komen deze variabelen resp. constanten namelijk altijd na elkaar voor. Overeenkomstig breiden we de notatie $b(x|d)$ uit tot $b(\bar{x}|\bar{d})$, en de notatie $\forall x\varphi$ tot $\forall\bar{x}\varphi$.

We laten twee dingen zien:

1. Als $\forall\bar{x}\exists y\varphi$ vervult wordt door de combinatie $\mathcal{M},b$ dan wordt $\forall\bar{x}[f(\bar{x})/y]\varphi$ nog steeds vervult, zij het door de combinatie $\mathcal{M}',b$, waarbij structuur $\mathcal{M}'$ een kleine wijziging is van structuur $\mathcal{M}$.

2. Als $\mathcal{M},b \models \forall\bar{x}[f(\bar{x})/y]\varphi$ voor een structuur $\mathcal{M}$ en bedeling b, dan vervult dezelfde structuur en bedeling ook $\forall\bar{x}\exists y\varphi$.

(2) $\Rightarrow$ (1): veronderstel dat $\mathcal{M},b \models \forall\bar{x}[f(\bar{x})/y]\varphi$ voor een structuur $\mathcal{M}$ en bedeling b:

$$\begin{aligned} &\mathcal{M},b \models \forall\bar{x}[f(\bar{x})/y]\varphi \\ \Leftrightarrow\ &\text{voor alle } \bar{d}\in D^n : \mathcal{M},b(\bar{x}|\bar{d}) \models [f(\bar{x})/y]\varphi \end{aligned}$$

Vanwege het substitutielemma mogen we nu de substitutie $f(\bar{x})/y$ omzetten naar een semantische substitutie $y|d$ op de bedeling $b(\bar{x}|\bar{d})$, waarbij

$$\begin{aligned} d = W_{\mathcal{M},b(\bar{x}|\bar{d})}(f(\bar{x})) = \\ (f)_I(W_{\mathcal{M},b(\bar{x}|\bar{d})}(\bar{x})) = (f)_I(b(\bar{x}|\bar{d})(\bar{x})) = (f)_I(\bar{d}) : \end{aligned}$$

$$\begin{aligned} \Leftrightarrow\ &\text{voor alle } \bar{d}\in D^n : d = (f)_I(\bar{d}) \text{ en } \mathcal{M},b(\bar{x}|\bar{d})(y|d) \models \varphi \\ \Rightarrow\ &\text{voor alle } \bar{d}\in D^n \text{ is er een } d\in D : \mathcal{M},b(\bar{x}|\bar{d})(y|d) \models \varphi \\ \Leftrightarrow\ &\text{voor alle } \bar{d}\in D^n : \mathcal{M},b(\bar{x}|\bar{d}) \models \exists y\varphi \\ \Leftrightarrow\ &\mathcal{M},b \models \forall\bar{x}\exists y\varphi. \end{aligned}$$

(1) $\Rightarrow$ (2): vooronderstel dat, voor een willekeurig paar $\mathcal{M},b$: $\mathcal{M},b \models \forall\bar{x}\exists y\varphi$.

$$\begin{aligned} &\mathcal{M},b \models \forall\bar{x}\exists y\varphi \\ \Leftrightarrow\ &\text{voor alle } \bar{d}\in D^n : \mathcal{M},b(\bar{x}|\bar{d}) \models \exists y\varphi \\ \Leftrightarrow\ &\text{voor alle } \bar{d}\in D^n \text{ is er een } d\in D : \mathcal{M},b(\bar{x}|\bar{d})(y|d) \models \varphi \ (*) \end{aligned}$$

Gegeven (*) kunnen we een functie $F : D^n \to D$ definiëren zodanig dat

$$d =_{Def} F(\bar{d})$$

en F dus voldoet aan (*). Als er meerdere zulke d's in D bestaan, dan kiezen we een willekeurige d. (We gebruiken hier dus weer impliciet het keuzeaxioma.) Laat vervolgens f een nieuw functiesymbool zijn dat nog niet voorkwam in de lijst van functiesymbolen van $\mathcal{M}$ en definieer $\mathcal{M}'$ als $\mathcal{M}$ plus het nieuwe functiesymbool f. Interpreteer vervolgens f als F. Dus: $(f)_I =_{Def} F$.

$$\begin{aligned} &\text{voor alle } \bar{d}\in D^n \text{ is er een } d\in D : \mathcal{M},b(\bar{x}|\bar{d})(y|d) \models \varphi \\ \Leftrightarrow\ &\text{voor alle } \bar{d}\in D^n \text{ is er een } d\in D : \mathcal{M}',b(\bar{x}|\bar{d})(y|d) \models \varphi \\ \Rightarrow\ &\text{voor alle } \bar{d}\in D^n : \mathcal{M}',b(\bar{x}|\bar{d})(y|(f)_I(\bar{d})) \models \varphi \\ &\qquad\vdots \\ \Leftrightarrow\ &\text{voor alle } \bar{d}\in D^n : \mathcal{M}',b(\bar{x}|\bar{d}) \models [f(\bar{x})/y]\varphi \\ \Leftrightarrow\ &\mathcal{M}',b \models \forall\bar{x}[f(\bar{x})/y]\varphi. \end{aligned}$$

Waarmee bewezen is dat $\forall\bar{x}[f(\bar{x})/y]\varphi$ ook vervulbaar is. □

Blz. 156, it. 1: $\forall xQxf(x)$.

Blz. 156, it. 2: Qab.

Blz. 156, it. 3: $\forall z(Qab \wedge Rbz)$.

Blz. 156, it. 4: $\forall z(Qf(z)g(z) \wedge Rg(z)z)$.

Blz. 156, it. 5:
$\forall x\forall y(Rxyz(x,y) \to (Pyz(x,y)u(x,y) \to Qf(u(x,y)z(x,y))))$

Blz. 156, it. 6:
$\forall x\forall y(Rg(x,y)h(x,y) \to (Pg(x,y) \to Qf(h(x,y))))$

Blz. 161, it. 1e:

1.	$\neg\exists Px$	ass
2.	$\quad Pa$	ass
3.	$\quad \exists xPx$	I$\exists$, 2
4.	$\quad \bot$	G$\neg$, 1, 3
5.	$\neg Pa$	I$\neg$
6.	$\forall x\neg Px$	I$\forall$, 5

Toelichting: om $\neg Pa$ te bewijzen voor willekeurige a, beginnen we het tegendeel aan te nemen, te weten Pa. In regel (5) hangt deze a niet meer af van eerdere aannames, en mogen we deze formule daarom veralgemeniseren.

Blz. 161, it. 1f: Als in de vorige uitwerking, alleen dan $\neg Pa$ aannemen.

1.	$\neg\exists\neg Px$	ass
2.	$\quad \neg Pa$	ass
3.	$\quad \exists x\neg Px$	I$\exists$, 2
4.	$\quad \bot$	G$\neg$, 1, 3
5.	$\neg\neg Pa$	I$\neg$
6.	Pa	$\neg\neg$-regel, 5
7.	$\forall xPx$	I$\forall$, 6

Blz. 161, it. 1g: Dit is een goede oefening om de G∃-regel te leren te beheersen. G∃-regel twee keer toepassen.

1.	$\exists x\exists y Pxy$	ass
2.	$\quad\exists y Pay$	ass
3.	$\quad\quad Pab$	ass
4.	$\quad\quad\exists x Pxb$	I∃, 3
5.	$\quad\quad\exists y\exists x Pxy$	I∃, 4
6.	$\quad Pab \to \exists y\exists x Pxy$	I→
7.	$\quad\exists y\exists x Pxy$	G∃, 2, 6
8.	$\exists y Pay \to \exists y\exists x Pxy$	I→
9.	$\exists y\exists x Pxy$	G∃, 1, 8

Toelichting: in de stap van regel (1) naar regel (2) mogen we niet *direct* Pab neerzetten, omdat de G∃-regel altijd maar kan worden toegepast op slechts één variabele (cf. 7, 9).

Blz. 161, it. 1h:

1.	$\exists y\forall x Pxy$	ass
2.	$\quad\forall x Pxb$	ass
3.	$\quad Pab$	G∀, 2
4.	$\quad\exists y Pay$	I∃, 3
5.	$\forall x Pxb \to \exists y Pay$	I→
6.	$\exists y Pay$	G∃, 1, 5
7.	$\forall x\exists y Pxy$	I∀, 6

Merk op dat hier alle vier de regels I∃, G∀, I∀ en G∃ worden gebruikt.

Blz. 161, it. 1i: Formeel analogon van bewijs uit het ongerijmde (*reductio ad absurdum*):

1.	$\neg\forall x\neg Px$	ass
2.	$\quad\neg\exists x Px$	ass
3.	$\quad\quad Pa$	ass
4.	$\quad\quad\exists x Px$	I∃, 3
5.	$\quad\quad\bot$	G¬, 2, 4
6.	$\quad\neg Pa$	I¬
7.	$\quad\forall x\neg Px$	I∀, 6
8.	$\quad\bot$	G¬, 1, 7
9.	$\neg\neg\exists x Px$	I¬
10.	$\exists x Px$	¬¬-regel, 9

Toelichting: in regel (2) wordt gezegd: stel niet. Om nu tot een tegenspraak (8) te komen, proberen we met een willekeurige constante, a, $\forall x\neg Px$ af te leiden. In (3) zeggen we weer: stel niet.

Blz. 161, it. 2a:

1.	$\exists x(Px \wedge Qx)$	ass
2.	$\quad Pa \wedge Qa$	ass
3.	$\quad Pa$	G∧, 2
4.	$\quad Qa$	G∧, 2
5.	$\quad\exists x Px$	I∃, 3
6.	$\quad\exists y Qy$	I∃, 4
7.	$\quad\exists x Px \wedge \exists y Qy$	I∧, 5, 6
8.	$(Pa \wedge Qa) \to (\exists x Px \wedge \exists y Qy)$	I→
9.	$\exists x Px \wedge \exists y Qy$	G∃, 1, 8

Toelichting: in (2) stellen we $\exists x(Px \wedge Qx)$ voor a willekeurig. (Standaard gebruik van existentiële formule.) Regels (2-4) is "uitpellen," regels (5-7) is samenstellen. Centrale punt in het bewijs is regel (9): toepassing van G∃.

Blz. 161, it. 2b:

1.	$\forall x(Px \to Qx)$	
2.	$\exists x Px$	ass
3.	$\quad Pa$	ass
4.	$\quad Pa \to Qa$	G∀, 1
5.	$\quad Qa$	G→, 3, 4
6.	$\quad\exists x Qx$	I∃, 5
7.	$Pa \to \exists x Qx$	I→
8.	$\exists x Qx$	G∃, 2, 7

Toelichting: merk op hoe de existentiële formule (2) weer standaard gebruikt wordt: als $\exists x Px$ (2), stel dan eens dat a die x is (3). Vervolgens kom je bij (7) uit op het verlangde resultaat. G∃, 2 en 7 geven dan $\exists x Qx$ onvoorwaardelijk.

Blz. 161, it. 2c:

1.	$\exists x\neg Px$	ass
2.	$\quad\neg Pa$	ass
3.	$\quad\quad\forall x Px$	ass
4.	$\quad\quad Pa$	G∀, 3
5.	$\quad\quad\bot$	G¬, 2, 4
6.	$\quad\neg\forall x Px$	I¬
7.	$\neg Pa \to \neg\forall x Px$	I→
8.	$\neg\forall x Px$	G∃, 1, 7

Toelichting: in regel (2) wordt op de standaard manier G∃ toegepast. Vervolgens wordt, om een tegenspraak te forceren, op regel (3) $\forall x Px$ aangenomen.

Blz. 161, it. 2d:

1.	$\forall x\neg Px$	ass
2.	$\quad\exists xPx$	ass
3.	$\quad\quad Pa$	ass
4.	$\quad\quad \neg Pa$	G$\forall$, 1
5.	$\quad\quad \bot$	G$\neg$, 3, 4
6.	$\quad Pa \to \bot$	I$\to$
7.	$\quad \bot$	G$\exists$, 2, 6
8.	$\neg\exists xPx$	I$\neg$

Toelichting: ook dit is weer een bewijs uit het ongerijmde. Eerst wordt in (2) het tegengestelde aangenomen, met de bedoeling op een tegenspraak (7) uit te komen. Bij (6) hadden we de I$\neg$-regel kunnen toepassen, maar dat hebben we niet gedaan want anders kwamen we bij $\neg Pa$ terecht wat niet direct leidt tot $\neg\exists xPx$. Dus passen we I$\to$ toe. De toepassing van de G$\exists$-regel op (7) is wat typisch (maar verder correct).

Blz. 161, it. 2e:

1.	$\neg\forall xPx$	ass
2.	$\quad\neg\exists x\neg Px$	ass
3.	$\quad\quad \neg Pa$	ass
4.	$\quad\quad \exists x\neg Px$	I$\exists$, 3
5.	$\quad\quad \bot$	G$\neg$, 2, 4
6.	$\quad \neg\neg Pa$	I$\neg$
7.	$\quad Pa$	$\neg\neg$-regel, 6
8.	$\quad \forall xPx$	I$\forall$, 7
9.	$\quad \bot$	G$\neg$, 1, 8
10.	$\neg\neg\exists x\neg Px$	I$\neg$
11.	$\exists x\neg Px$	$\neg\neg$-regel, 10

Toelichting: Net als een eerdere opgave is dit ook een bewijs uit het ongerijmde. In regel (2) zeggen we: “stel niet”. Om een tegenspraak te creëren proberen dan vervolgens $\forall xPx$ te bewijzen. Dit doen we door Pa te bewijzen voor willekeurige a. Op regel (7) blijkt dat a inderdaad willekeurig is.

Blz. 161, it. 2f:

1.	$\forall x(Px \to Qx)$	
2.	$\exists x\neg Qx$	ass
3.	$\quad\neg Qa$	ass
4.	$\quad\quad Pa$	ass
5.	$\quad\quad Pa \to Qa$	G$\forall$, 1
6.	$\quad\quad Qa$	G$\to$, 4, 5
7.	$\quad\quad \bot$	G$\neg$, 3, 6
8.	$\quad \neg Pa$	I$\neg$
9.	$\quad \exists x\neg Px$	I$\exists$, 8
10.	$\neg Qa \to \exists x\neg Px$	I$\to$
11.	$\exists x\neg Px$	G$\exists$, 2, 10

Blz. 161, it. 2g:

1.	$\forall x(Px \vee Qx)$	
2.	$\exists x\neg Qx$	ass
3.	$\quad\neg Qa$	ass
4.	$\quad Pa \vee Qa$	G$\forall$, 1
5.	$\quad\quad Pa$	ass
6.	$\quad\quad Pa$	Herh, 5
7.	$\quad Pa \to Pa$	I$\to$
8.	$\quad\quad Qa$	ass
9.	$\quad\quad \bot$	G$\neg$, 3, 8
10.	$\quad Qa \to Pa$	I$\to$
11.	$\quad Pa$	G$\vee$, 4, 7, 10
12.	$\quad \exists xPx$	I$\exists$, 11
13.	$\neg Qa \to \exists xPx$	I$\to$
14.	$\exists xPx$	G$\exists$, 2, 13

Toelichting: een lastige opgave, omdat het bijna onvermijdelijk lijkt dat zowel de G$\vee$- als de G$\exists$-regel moeten worden toegepast. (Wij hebben in ieder geval geen bewijs kunnen vinden zonder dat één van deze twee regels worden gebruikt.)

Blz. 161, it. 2h:

1.	$\forall x(Px \to Qx)$	
2.	$\exists x(Px \wedge Rx)$	ass
3.	$Pa \wedge Ra$	ass
4.	Pa	G$\wedge$, 3
5.	Ra	G$\wedge$, 3
6.	$Pa \to Qa$	G$\forall$, 1
7.	Qa	G$\to$, 4, 6
8.	$Qa \wedge Ra$	I$\wedge$, 5, 7
9.	$\exists x(Qx \wedge Rx)$	I$\exists$, 8
10.	$(Pa \wedge Ra) \to \exists x(Qx \wedge Rx)$	I$\to$
11.	$\exists x(Qx \wedge Rx)$	G$\exists$, 2, 10

Blz. 162, it. 14.12: x voor y substitueren in $\neg\forall x\ x = y$ levert $\neg\forall x\ x = x$, en dat is een contradictie (niet vervulbaar), want elk object is gelijk aan zichzelf.

$\forall y\neg\forall x\ x = y$ is equivalent met $\forall y\exists x\ \neg x = y$; deze formule beschrijft de situatie dat er minstens twee objecten zijn; de voorzin van de implicatie is dus vervulbaar. Kortom, elk model met meer dan één object vormt een tegenvoorbeeld met betrekking tot de voorlopige versie van axioma 4.

Blz. 162, it. 14.13: 1. nee; substitutie zou geven: $Px \wedge \exists xRxx$ en daarin is de laatste x niet meer vrij waar y eerst wel vrij was.

2. ja; het substitutie-resultaat is $Px \wedge \exists zRzx$ en x is vrij waar y vrij was.

3. ja; de substitutie verandert de formule niet.

4. nee; het eerste voorkomen van y is voor substitutie vrij, maar erna staat daar de gebonden variabele x.

5. ja; de substitutie verandert de formule niet.

6. ja; op de plaats van de vrije y staat ook na substitutie een vrije variabele (x).

Blz. 163, it. 14.14:

1	$\forall x(Ax \to Bx)$	[aanname]
2	$\forall x(Ax \to Bx) \to (Aa \to Ba)$	[axioma 4]
3	$Aa \to Ba$	[MP uit 1 en 2]
4	Aa	[aanname]
5	Ba	[MP uit 3 en 4]

Blz. 163, it. 14.15:

1.	$\forall x(\neg Ax \to \neg Bx)$	[aanname]
2.	$\forall x(\neg Ax \to \neg Bx) \to (\neg Aa \to \neg Ba)$	[axioma 4]
3.	$\neg Aa \to \neg Ba$	[MP uit 1 en 2]
4.	$(\neg Aa \to \neg Ba) \to (Ba \to Aa)$	[axioma 3]
5.	$Ba \to Aa$	[MP uit 3 en 4]
6.	Ba	[aanname]
7.	Aa	[MP uit 5 en 6]
8.	$\forall x(Ax \to Cx)$	[aanname]
9.	$\forall x(Ax \to Cx) \to (Aa \to Ca)$	[axioma 4]
10.	$Aa \to Ca$	[MP uit 8 en 9]
11.	Ca	[MP uit 7 en 10]

Blz. 165, it. 14.16: Voor het bewijs is het handig uit te gaan van het principe:

$$\textit{lemma} \qquad T \vdash \varphi \iff \overline{T} \vdash \varphi$$

Bewijs van het lemma:

($\Leftarrow$) Omdat $T \subseteq \overline{T}$, geldt dat alle φ die afleidbaar zijn uit T ook afleidbaar zijn uit $\overline{T}$.

($\Rightarrow$) Stel dat φ uit $\overline{T}$ afleidbaar is. Laat de afleiding $\mathscr{A}$ van φ gebruik maken van premissen $\psi_1,\ldots,\psi_n$ uit $\overline{T}$. Elke ψ_i is dus per definitie afleidbaar uit T. Dan kunnen we φ ook direct uit T afleiden door eerst de afleidingen van $\psi_1,\ldots,\psi_n$ uit T onder elkaar te zetten en dan te vervolgen met afleiding $\mathscr{A}$. □

Nu de eerste bewering: als T consistent is, dan $\overline{T}$ ook.

Bewijs van de bewering: Neem aan dat T consistent is, maar $\overline{T}$ niet. Dan is er een φ waarvoor zowel $\overline{T} \vdash \varphi$ als $\overline{T} \vdash \neg\varphi$. Nu volgt met het lemma dat dan ook $T \vdash \varphi$ en $T \vdash \neg\varphi$, wat in tegenspraak is met de veronderstelde consistentie van T. □

En omgekeerd: Stel dat $\overline{T}$ consistent is, maar T niet. Dan is er een φ zodat zowel $T \vdash \varphi$ als $T \vdash \neg\varphi$; weer met gebruik van het lemma uit het vorige antwoord volgt dat dan $\overline{T} \vdash \varphi$ en $\overline{T} \vdash \neg\varphi$, hetgeen in strijd is met de aanname dat $\overline{T}$ consistent is. □

Blz. 166, it. 14.17: We doen inductie naar de opbouw van φ.

- *Basisstap*: als φ een atomaire formule is en b en b' kennen dezelfde waarden toe aan de vrije variabelen in φ dan zal de waardefunctie W dezelfde waarden toekennen aan alle termen in φ, met als gevolg dat $V_b(\varphi) = V_{b'}(\varphi)$.
- *Inductiestap*: stel voor elke echte deelformule ψ van φ geldt dat als $b(v) = b'(v)$ voor elke vrije variable v in ψ, dan $V_b(\psi) = V_{b'}(\psi)$ (inductiehypothese).
  Laat nu b en b' dezelfde waarden geven aan de vrije variabelen in φ. Als φ de vorm $\neg\psi$ heeft, dan volgt hieruit onmiddellijk met behulp van de waarheidsdefinitie voor de connectieven dat $V_b(\varphi) = V_{b'}(\varphi)$.
  Als $\varphi = \psi \to \chi$, dan is de inductiehypothese van toepassing op ψ en χ, terwijl V_b en $V_{b'}$ gelijk zijn voor de vrije variabelen in ψ en χ, want die zijn ook vrij in φ; de waarheidstabel voor $\to$ doet de rest.

Als φ de vorm $\forall v\psi$ heeft dan heeft ψ mogelijkerwijs de variabele v vrij. Merk echter op dat desalniettemin uit de inductiehypothese volgt dat $b(v|d)$ en $b'(v|d)$ voor elke vrije variabele in ψ dezelfde waarde opleveren. Hieruit volgt met behulp van de waarheidsdefinitie voor de universele kwantor dat $V_b(\varphi) = V_{b'}(\varphi)$. □

Blz. 169, it. 14.18: (⇒) Bij opdracht 414 is al bewezen dat $\overline{T}$ consistent is als T dat is. $\overline{T}$ is ook *maximaal* consistent, want stel $\varphi \notin \overline{T}$, dan $T \nvdash \varphi$, dus (T is volledig) $T \vdash \neg\varphi$. Maar dan is $\overline{T} \cup \{\varphi\}$ inconsistent.

(⇐) $\overline{T}$ is consistent dus ook T (zie opdracht 414). T is ook volledig want stel $T \nvdash \varphi$, dan $\varphi \notin \overline{T}$, dus (stelling 9.11) $\neg\varphi \in \overline{T}$, ergo $T \vdash \neg\varphi$. □

Blz. 170, it. 14.19: De zin kan, gegeven zijn aanwezigheid in dit boek niet onwaar zijn! Stel maar dat de zin toch onwaar is. Dan is het enerzijds zo dat er onwaarheden ("fouten") in dit boek staan, anderzijds vanwege de inhoud van de zin, dat er juist géén fouten instaan. Dit is onmogelijk (als we tenminste 'fout' en 'onwaarheid' gelijkstellen, en de zin echt *gebruiken* en niet slechts *noemen*). De zin kan echter best waar zijn; we kunnen hem bijvoorbeeld vervullen door hem in een boek te plaatsen ...

Blz. 170, it. 14.20: Stel de zin is waar. Dan geldt dus, dat zegt de zin immers, dat ze onwaar is, hetgeen in tegenspraak is met de aanname.

Stel nu dat de zin onwaar is. Dat is ook wat de zin beweert; de zin is dus waar, hetgeen weer in tegenspraak is met de aanname.

Blz. 170, it. 14.21: De 'leugenaarszin' X is waar desda $\mathtt{NA}X$ waar is. Verder weten we dat voor elk rijtje Y geldt dat $\mathtt{VNA}Y$ waar is mits $\mathtt{NA}YY$ dat is. Kiezen we dus $X = \mathtt{VNA}Y$ dan zitten we al een stuk in de goede richting: dan $\mathtt{NA\ VNA}Y$ (de spatie dient slechts de leesbaarheid) $\Leftrightarrow$ $\mathtt{NA}X \Leftrightarrow X \Leftrightarrow \mathtt{VNA}Y \Leftrightarrow \mathtt{NA}YY$. Maar de "vergelijking" $\mathtt{NA\ VNA}Y \Leftrightarrow \mathtt{NA}YY$ heeft natuurlijk een oplossing: $Y = \mathtt{VNA}$. En dus is $X = \mathtt{VNA\ VNA}$ de gevraagde leugenaarszin.

Deze X is waar, want stel dat $\mathtt{VNA\ VNA}$ afdrukbaar is, dan zegt de correctheid van de machine dat X dan waar moet zijn; maar dan is $\mathtt{VNA\ VNA}$ juist *niet* afdrukbaar! Tegenspraak; dus X is niet afdrukbaar en daarmee waar. We kunnen, desnoods met brute kracht, nagaan dat dit ook het simpelste ware tekenrijtje is dat van zichzelf zegt dat het niet afdrukbaar is.

Blz. 171, it. 15.1: 1. $\exists_t x Zkx \wedge \neg\forall_t x Zkx$

discussiedomein met soorten	*vertaalsleutel*
• m : mensen	• Zxy : x is ziek op y
• t : tijdstippen	• k : Kortjakje

2. $\forall_d x(Zx \to \exists_b y Gkyx)$

discussiedomein met soorten	*vertaalsleutel*
• d : dagen van de week	• $Gxyz$: x gaat naar y op z
• b : kerken (bidplaatsen)	• Zx : x is een zondag
• m : mensen	• k : Kortjakje

3. $\forall_m x \exists_t y Zxy$

discussiedomein met soorten	*vertaalsleutel*
• m : mensen	• Zxy : x is ziek op y
• t : tijdstippen	

Blz. 172, it. 15.2: 1. $\exists t Zkt \wedge \neg\forall t Zkt$

discussiedomein met soorten	*vertaalsleutel*
• m : mensen	• $Z_{\langle m,t\rangle}xy$: x is ziek op y
• t : tijdstippen	• k_m : Kortjakje

2. $\forall w(Zw \to \exists y Gkyw)$

discussiedomein met soorten	*vertaalsleutel*
• d : weekdagen	• $G_{\langle m,b,d\rangle}xyz$: x gaat naar y o
• b : kerken	• $Z_{\langle d\rangle}x$: x is een zondag
• m : mensen	• k_m : Kortjakje

w is hier een variabele van soort d, en y van soort b.

3. $\forall x \exists t Zxt$

discussiedomein met soorten	*vertaalsleutel*
• m : mensen	• $Z_{\langle m,t\rangle}xy$: x is ziek op y
• t : tijdstippen	

x is hier een variabele van soort m.

Blz. 172, it. 15.3: – de vertaling van $\forall_m x \exists_t y Oxy \wedge \exists_m z \forall_t u Ozu$ staat al in de tekst

– $\exists_m x \exists_t y Oxy \rightsquigarrow \exists x(Mx \wedge \exists y(Ty \wedge Oxy))$

– $\forall_t x Zkx \rightsquigarrow \forall x(Tx \to Zkx)$

– $\exists_m x \forall_t y(k = x \wedge Zxy) \rightsquigarrow \exists x(Mx \wedge \forall y(Ty \to (k = x \wedge Zxy)))$

Blz. 173, it. 15.4: 1. $\neg\forall x{\in}M : Ax$
discussiedomein: mensen
vertaalsleutel:

– Ax : x is aardig;
– Mx : x is een mens.

2. $\exists x{\in}V\, \forall y{\in}M : Nxy$
discussiedomein: mensen
vertaalsleutel:

– Mx : x is een man;
– Nxy: x minacht y;
– Vx : x is een vrouw.

3. $\neg\exists x{\in}M\, \exists y{\in}K : Wxy \to \neg\exists x{\in}M\, \exists y{\in}K : Vxy$
discussiedomein: mensen en kenbare feiten
vertaalsleutel:

– Kx : x is een kenbaar feit;

– Mx : x is een mens;
– Vxy : x verklapt y;
– Wxy : x weet y.

4. $\forall x{\in}M\,\exists y{\in}M : Hxy \wedge \exists x{\in}M\,\forall y{\in}M : Hxy$
discussiedomein: mensen
vertaalsleutel:

– Hxy : x houdt van y;
– Mx : x is een mens.

Blz. 173, it. 15.5: $\forall X(\forall x((Dx \wedge Vx) \to Xx) \to Xp) \wedge \neg\exists X(\forall x((Sx \wedge Lx) \to Xx) \wedge Xp)$ waarin:

– Dx : x is een dictator
– Lx : x is geliefd
– Sx : x is een staatshoofd
– Vx : x is gevreesd
– p : Pinochet

Blz. 174, it. 15.6: $\exists x(\forall y(Ky \leftrightarrow \forall X(Xy \to Xx)) \wedge Tx)$ waarin:

– Kx : x is een koning
– Tx : x toornt

Blz. 174, it. 15.7: Het is duidelijk dat $\forall X(Xx \leftrightarrow Xy)$ impliceert dat $\forall X(Xx \to Xy)$ Het omgekeerde volgt natuurlijk als we kunnen bewijzen dat $\forall X(Xx \to Xy) \Rightarrow \forall X(Xy \to Xx)$. Er zijn verschillende manieren om dit aan te tonen:

– (met Leibniz' principe) $\forall X(Xx \to Xy) \Leftrightarrow x = y \Leftrightarrow y = x \Leftrightarrow \forall X(Xy \to Xx)$. Dit eenvoudige bewijs kunnen we echter niet gebruiken wanneer we '=' willen *definiëren* in tweede orde logica.
– (met stof uit § 15.4) Laat $\forall X(Xx \to Xy)$ gegeven zijn. Kies nu een willekeurige predikaatvariabele Y. Dan leidt de gegeven formule voor de instantiatie $X = \lambda z[\neg Yz]$ tot: $\neg Yx \to \neg Yy$. Met contrapositie volgt hieruit $Yy \to Yx$, en dus in het algemeen $\forall Y(Yy \to Yx)$, ofwel $\forall X(Xy \to Xx)$.
– (met stof uit § 15.5) Een semantische tegenhanger van het vorige semi-deductieve bewijsje.

Blz. 174, it. 15.8:

$$\{\langle A,B\rangle \mid A \subseteq D \text{ en } B \subseteq D \text{ en } |A \cap B| \leq 5\}$$

Blz. 174, it. 15.9:

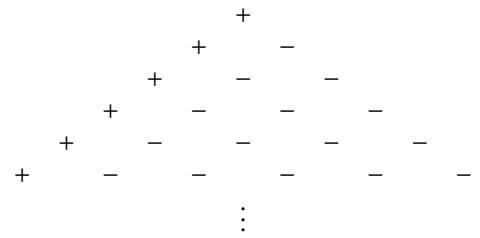
$$\{\langle A,B\rangle \mid A \subseteq D \text{ en } B \subseteq D \text{ en } A \cap B = \emptyset\}$$

Blz. 174, it. 15.10:

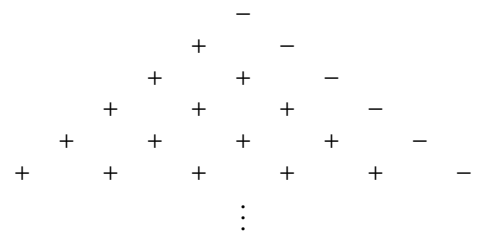
$$\{\langle A,B\rangle \mid A \subseteq D \text{ en } B \subseteq D \text{ en } A \not\subseteq B\}$$

Blz. 176, it. 15.11:

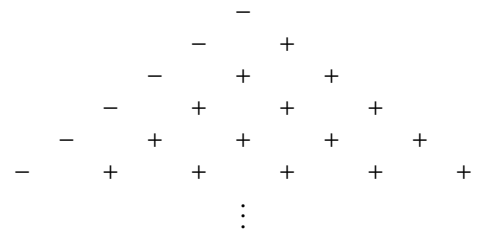
1. **geen**:



2. **niet alle**:



3. **minstens één** (of gewoon **een**):

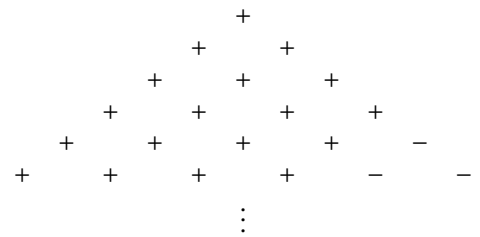


4.

– **geen** en **alle** zijn elkaars spiegelbeeld ten opzichte van de verticale as die de piramide middendoor deelt.
– **geen** en **minstens één** zijn elkaars complement (dat wil zeggen alle plussen veranderen in minnen en alle minnen in plussen).
– **geen** en **niet alle** zijn het spiegelbeeld van elkaars complement (of omgekeerd, het complement van elkaars spiegelbeeld).
– **niet alle** en **alle** zijn elkaars complement.
– **niet alle** en **minstens één** zijn elkaars spiegelbeeld.
– **minstens één** en **alle** zijn het spiegelbeeld van elkaars complement (of omgekeerd, het complement van elkaars spiegelbeeld).

Blz. 176, it. 15.12:

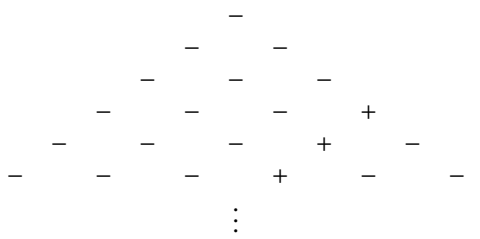
1. **hoogstens drie**:



2. In dit patroon staat een '+' op elke plaats waar in het patroon voor **minstens drie** een '–' staat en bovendien

overal op de linker-diagonaal door het $\langle 0,3\rangle$. Die diagonaal bestaat in het patroon voor **hoogstens drie** ook uit plussen, en is daar de 'grenslijn' van de driehoek met plussen.

3. **precies drie**:

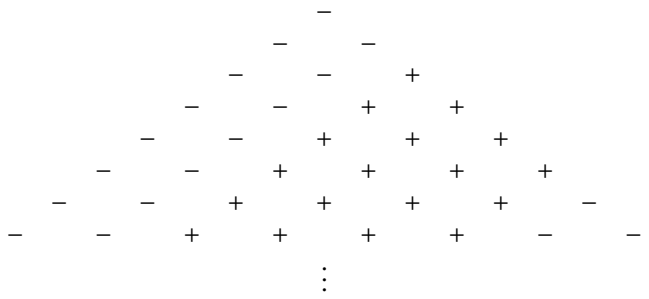


4. In dit laatste patroon staat alleen op de grenslijn van beide andere patronen een '+'. Algemener: daar waar de patronen voor **minstens drie** en **hoogstens drie** beide een '+' hebben.

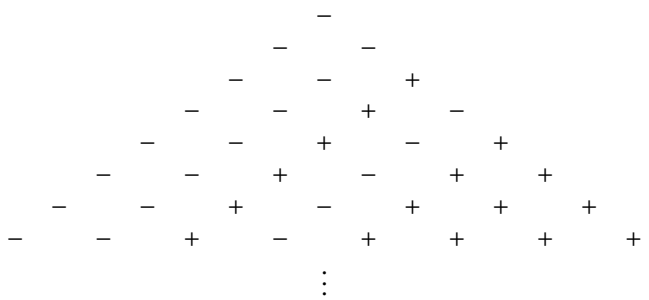
5. **Precies drie** kan immers geparafraseerd worden als **hoogstens drie en minstens drie**. Algemener: *conjunctie* van twee kwantoren levert een piramide-patroon op met plussen op slechts die plaatsen die in beide patronen van de samenstellende kwantoren een plus hadden, kortom de *doorsnijding* van de plus-patronen. Evenzo levert *disjunctie* van kwantoren een piramide-patroon op met plussen op de plaatsen die in minstens één van beide patronen van de samenstellende kwantoren een plus hadden; dat wil zeggen de *vereniging* van de plus-patronen.

Blz. 176, it. 15.13:

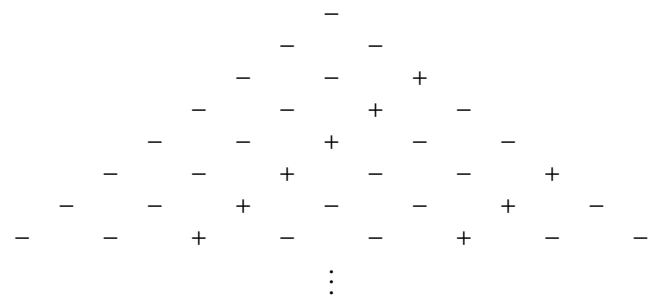
1. **minstens twee en hoogstens vijf**:



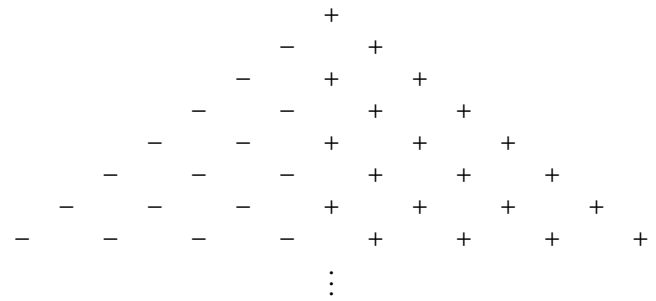
2. **precies twee of minstens vier**:



3. **precies twee of precies vijf**:



4. **minstens de helft**:



Blz. 177, it. 15.14:

1. Niet reflexief. *geen A zijn A* is alleen dan waar als $A = \emptyset$. In de regel zal dit geval zich niet voordoen. In een domein met minstens één mens geldt *geen mensen zijn mensen* niet.

2. Ook niet irreflexief. *geen A zijn A* is niet altijd onwaar. In een domein zonder weerwolven geldt *geen weerwolven zijn weerwolven* wél.

3. Symmetrisch. *geen voetballers zijn hackers* impliceert dat *geen hackers zijn voetballers*. Dit is conform de gegeven interpretatie van *geen*.

4. Niet antisymmetrisch. De voetballers uit het vorige onderdeel hoeven immers niet de hackers te zijn.

5. Niet transitief. Uit *geen hackers zijn voetballers* en *geen voetballer zijn hackers* volgt immers niet dat *geen hackers zijn hackers*.

Blz. 177, it. 15.15:

1. Niet reflexief. *precies twee A zijn A* is waar desda $|A| = 2$; *precies twee voetballers zijn voetballers* is dus onwaar als er meer of juist minder dan twee voetballers zijn.

2. Niet irreflexief. Als er exact twee voetballers in het domein zijn dan is bovenstaande zin juist wel waar.

3. Symmetrisch. Het volgt uit *precies twee voetballers zijn hackers* dat *precies twee hackers zijn voetballers*.

4. Antisymmetrisch. Opnieuw: de voetballers in het vorige voorbeeld hoeven natuurlijk niet precies de hackers te zijn.

5. Niet transitief. Stel maar voor: twee voetballende hackers en een niet-voetballende hacker. Dan is *precies twee H V* waar en *precies twee V H* ook, maar *precies twee H H* niet, want dan $|H| = 3$.

Blz. 177, it. 15.16: 1. We nemen aan dat $\mathcal{Q}$ de conservativiteits- en kwantiteits-eigenschap bezit. Dat $\mathcal{Q}$ reflexief is, wil zeggen: $\forall X \mathcal{Q} X X$. Dit betekent dan dat, hoe we X ook kiezen, op plaats $\langle |X - X|, |X \cap X| \rangle$ in de piramide een '+' moet staan. Als $n = |X|$, dan heeft $\langle 0, n \rangle$ een '+'. De rechterzijde (dat wil zeggen de rechterdiagonaal door $\langle 0, 0 \rangle$) moet dus uit plussen bestaan.

2. Laat de rechterzijde van de piramide uit plussen bestaan en stel dat de kwantor $\mathcal{Q}$ niet reflexief is. Dan is er een verzameling A zo dat niet $\mathcal{Q}AA$. Dan zou er op plaats $\langle 0, |A| \rangle$ in de piramide geen plus staan.

Blz. 177, it. 15.17: $\mathcal{Q}$ is irreflexief. Immers stel van niet, dan is er een A is zodat $\mathcal{Q}AA$. Dan zou er op $\langle 0, |A| \rangle$ in de piramide een plus staan, quod non.

Blz. 178, it. 15.18:

$$\frac{\text{Geen kind is stout}}{\text{Geen kind is druk en stout}}$$

$$\frac{\text{Weinig kinderen aten}}{\text{Weinig kinderen aten veel}}$$

Blz. 178, it. 15.19: De verbale constituent mag het positief polaire *allerminst* bevatten **tenzij** het subject rechts neerwaarts monotoon (= rechts-dalend) is.

Blz. 178, it. 15.20:

Opwaartse monotonie links (linker stijging) van een kwantor:

Voor alle X, X', Y: als $\mathcal{Q}XY$ en $X \subseteq X'$ dan $\mathcal{Q}X'Y$.

Neerwaartse monotonie links (linker daling) van een kwantor:

Voor alle X, X', Y: als $\mathcal{Q}XY$ en $X' \subseteq X$ dan $\mathcal{Q}X'Y$.

1. links-dalend;
2. links-stijgend;
3. links-stijgend;
4. links-dalend;
5. noch links-stijgend, noch links-dalend;
6. links-dalend.

Blz. 180, it. 15.21:

1. $\lambda x[Px]$;
2. $\lambda x[Px](j)$ en dit levert Pj;
3. $\lambda Z[\exists x(Px \wedge Zx)]$;
4. $\lambda Z[\exists x(Yx \wedge Zx)]$;
5. Omdat Z niet vrij is voor Y, moet de formule eerst veranderd worden door voor Z een andere letter te nemen, bijvoorbeeld X. De formule wordt dan: $\lambda Y[\lambda X[\exists x(Yx \wedge Xx)]](Z)$. Conversie levert vervolgens: $\lambda X[\exists x(Zx \wedge Xx)]$;
6. $\lambda Y[Yj]$;
7. $\lambda x[Rax](j) \Rightarrow$ (nogmaals λ-conversie) Raj;
8. $\lambda Y[Ya](\lambda y[Py]) \Rightarrow \lambda y[Py](a) \Rightarrow Pa$;
9. (na herhaald converteren) $(Pb \wedge Qb) \vee Bb$.

Blz. 180, it. 15.22: 1. $\mathbf{V}(\lambda x[Bxg])$;

2. $\mathbf{T}(\lambda x[\exists y Gxy])$;
3. $\mathbf{Z}(\lambda x[\neg \exists y Vxy])$;
4. $\mathbf{Z}(\lambda x[\neg Vxx])$;
5. $\neg \mathbf{Z}(\lambda x[\neg \forall y Vxy])$.

Blz. 181, it. 15.23: 1. de Bakoenin-biografie aan Cornelis geven;

2. door Adriaan aan Cornelis gegeven worden;
3. van Adriaan de Bakoenin-biografie krijgen.

Blz. 181, it. 15.24: 1. $\lambda x[\exists y Gxby]$;

2. $\lambda x[\exists y Gybx]$;
3. $\lambda x[\exists y Gxyc]$;
4. $\lambda x[\exists y Gayx]$;
5. $\lambda x[\forall z \exists y Gxyz]$.

Blz. 181, it. 15.25: $\mathbf{P}(\lambda x W(x, \exists y Bxy)$

vertaalsleutel: $W(x, p)$: x wil p; Bxy: x bereikt y; $\mathbf{P}(X)$: X is prijzenswaardig.

Blz. 182, it. 15.26: **V** en **S** zijn van het type $\langle \langle e, t \rangle, t \rangle$; **M** en **P** van type $\langle e, \langle \langle e, t \rangle, t \rangle \rangle$.

Blz. 182, it. 15.27: $\mathcal{S}$, de vertaling van *snel*, is van het type $\langle \langle e, t \rangle, \langle e, t \rangle \rangle$.

Blz. 183, it. 15.28: 1. $\lambda x[Gxbc]$ (van type $\langle e, t \rangle$);

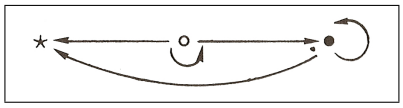
2. $\lambda y[\lambda x[Gxyc]]$ (van type $\langle e, \langle e, t \rangle \rangle$);
3. $\lambda x[Gcbx]$ (van type $\langle e, t \rangle$);
4. $\lambda x[\lambda y[Gcyx]]$ (van type $\langle e, \langle e, t \rangle \rangle$);
5. $\lambda y[\lambda z[\exists x Gxyz]]$ (van type $\langle e, \langle e, t \rangle \rangle$);
6. $\lambda x[\exists y Gxyc]$ (van type $\langle e, t \rangle$);
7. $\lambda z[\forall x \exists y Gxyz]$ (van type $\langle e, t \rangle$).

Blz. 183, it. 15.29:

1. de Bakoenin-biografie geven;
2. de Bakoenin-biografie krijgen;
3. iets geven;
4. iets krijgen.

De uitdrukkingen zijn alle van het type $\langle e, \langle e, t\rangle\rangle$.

Blz. 184, it. 15.30: Om de richting van de pijlen in het diagram hieronder te begrijpen, moet de lezer zich het volgende realiseren. Laat om de gedachten te bepalen $\star$, $\circ$ en $\bullet$ de interpretaties zijn van respectievelijk s, c en p. Dan geldt bijvoorbeeld $I(R)(\star)(\circ) = 1$, en daarmee $V(R(s)(c)) = 1$, oftewel in relationele notatie $V(Rcs) = 1$, en dus $\langle \circ, \star\rangle \in I(R)$: de pijl moet dus van $\circ$ naar $\star$ wijzen en niet omgekeerd!



Blz. 185, it. 15.31:

- Als P een n-plaatsige predikaatletter is en $t_1, \ldots, t_n$ zijn termen, dan is $Pt_1 \ldots t_n$ een welgevormde formule van de MPL;
- Als φ een welgevormde formule is van de MPL, dan $\neg\varphi$ ook.
- Als φ en ψ welgevormde formules zijn van de MPL, dan $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ en $(\varphi \leftrightarrow \psi)$ ook.
- Als φ een welgevormde formule is van de MPL, en v is een variabele, dan zijn $\forall v\varphi$ en $\exists v\varphi$ ook welgevormde formules van de MPL.
- Als φ een welgevormde formule is van de MPL, dan $\Box\varphi$ en $\Diamond\varphi$ ook.
- Niets anders is een welgevormde formule van de MPL.

Blz. 185, it. 15.32:

- $\exists x \Box Sx$ staat voor: 'er is iemand die noodzakelijkerwijs wint'.
- $\Box \exists x Sx$ staat voor: 'het is noodzakelijk dat er iemand wint'.

Blz. 185, it. 15.33: – 'Het is voor elke speler mogelijk dat hij verliest' $\rightsquigarrow \forall x \Diamond \neg Sx$;

- 'Er is een spelsituatie mogelijk waarbij elke speler verliest' $\rightsquigarrow \Diamond \forall x \neg Sx$.

Blz. 185, it. 15.34: Geen enkele formule is waar.

Blz. 186, it. 15.35: 1. Onwaar, want wereld 1 is toegankelijk voor zichzelf en in 1 geldt $\exists x Rxx$ niet.

2. Waar, want vanuit 2 is alleen 2 toegankelijk en daarin staat $\star$ tot zichzelf in relatie R.
3. Waar, want ook vanuit 3 is slechts 3 toegankelijk; tweemaal dit feit gebruiken en opmerken dat ook hier $\langle \star, \star\rangle \in I(R)$ volstaat.

Blz. 186, it. 15.36: We bewijzen dit uit het ongerijmde. Stel dus dat de te bewijzen bewering onjuist is. Dan is er een model $M = \langle D, W, T, I\rangle$ met T symmetrisch en een wereld $w \in W$ zodat voor een formule φ in dat model $\varphi \rightarrow \Box\Diamond\varphi$ onwaar is w, dus φ is waar in w en $\Box\Diamond\varphi$ onwaar. Uit het laatste volgt dat er een $v \in W$ bestaat waarvoor wTv en $\Diamond\varphi$ onwaar is in v; dit impliceert op zijn beurt dat voor elke u zodat vTu de formule φ onwaar is. Maar uit de symmetrie van T volgt w zelf zo'n u is, zodat φ in w onwaar zou zijn, wat in tegenspraak is met het eerder gestelde.

Blz. 186, it. 15.37: Als T *transitief* is, dan geldt het schema $\Box\varphi \rightarrow \Box\Box\varphi$.

We leveren voor de variatie een direct bewijs. Neem een willekeurig model $M = \langle D, W, T, I\rangle$ met T transitief en een wereld $w \in W$ zodat $\Box\varphi$ waar is in w. Kies vervolgens een willekeurige v met wTv; dan is φ waar in v. Beschouw nu een willekeurige u met vTu. Wegens de transitiviteit van T hebben we nu ook wTu, en dus is φ is waar in u. Derhalve is $\Box\varphi$ waar in v; en, omdat v een willekeurige T-opvolger is van w, $\Box\Box\varphi$ waar in w. Kortom $\Box\varphi \rightarrow \Box\Box\varphi$ is waar in w, en $\Box\varphi \rightarrow \Box\Box\varphi$ geldig in elk model met een transitieve T.

Blz. 193, it. 16.1: De feiten laten we over aan de gretige lezers van *Ons Koningshuis*. Hier zijn de regels voor *broer*, *zus*, *oom* en *tante*.

```
broer_van(X,Y) :-
      man(X),
      vader_van(V,X), vader_van(V,Y),
      moeder_van(M,X), moeder_van(M,Y),
      X\==Y.

zus_van(X,Y) :-
      vrouw(X),
      vader_van(V,X), vader_van(V,Y),
      moeder_van(M,X), moeder_van(M,Y),
      X\==Y.

oom_van(X,Y) :- ouder_van(O,Y), broer_van(X,O).

tante_van(X,Y) :- ouder_van(O,Y), zus_van(X,O).
```

De bewering `X\==Y` drukt uit dat `X` en `Y` niet letterlijk identiek zijn: dit is nodig om te coderen dat niemand een broer of zus van zichzelf is. Merk op dat `broer_van` en `zus_van` gevoelig zijn voor volgorde; de antwoorden

```
X = beatrix, Y = margriet
```

en

```
X = margriet, Y = beatrix
```

gelden dus als verschillend.

Blz. 193, it. 16.2: Hier is het gevraagde predikaat:

```
voorouder_van(X,Y) :- ouder_van(X,Y).
voorouder_van(X,Y) :- ouder_van(X,Z),
                      voorouder_van(Z,Y).
```

Blz. 193, it. 16.3: Vooraleer we aan het programmeren slaan is het zaak de strekking van de tekst te doorgronden. Vaart iedereen die een bebaarde man is mee? De eerste regel van het lied schijnt dit te suggereren. Of geldt voor iedereen die meevaart dat het een bebaarde man is? Deze suggestie wordt gewekt door de tweede regel. Wij houden het hier maar op het tweede. Dit leidt tot het volgende programma:

```
man(jan).
man(piet).
man(joris).
man(korneel).
man(klaas).

vrouw(marie).

bebaard(jan)
bebaard(piet).
bebaard(joris).
bebaard(korneel).

vaart_mee(X) :- man(X), bebaard(X).
```

De vraag naar wie er meevaart en het antwoord erop:

```
| ?- vaart_mee(X).

X = jan ;

X = piet ;

X = joris ;

X = korneel ;

no
| ?-
```

In feite wordt een PROLOG programma altijd zo gelezen dat de objecten die aan een definitie voldoen de enige objecten zijn die eraan voldoen. In dit voorbeeld: degenen die meevaren zijn *precies* de bebaarde mannen. Met andere woorden: als x een man met een baard is dan vaart x mee, en als x meevaart dan is x een man met een baard.

Blz. 194, it. 16.4: Het volgende PROLOG programma volgt de recursieve definitie van de faculteitsfunctie op de voet:

```
fac(0, s 0).
fac(s X, Y) :- fac(X, Z), maal(s X, Z, Y).
```

Blz. 195, it. 16.5: 1. Alle lijsten van minstens drie elementen met eerste element a en derde element b.

2. Alle lijsten met als eerste element een lijst van precies drie elementen waarvan het eerste een a is en het derde een b.
3. Alle lijsten met als eerste element de lege lijst.
4. Alle lijsten van twee elementen met als eerste element de lege lijst.
5. Alle lijsten van drie elementen met als derde element een a.

Blz. 195, it. 16.6: Het predikaat `xyz(X)` is waar desda `X` een lijst is met een even aantal elementen.

Blz. 195, it. 16.7: `zyx(X,Y,Z)` is waar desda `Z` een lijst is waarin `X` en `Y`—in die volgorde—voorkomen als elementen die elkaars buren zijn.

Blz. 196, it. 16.8:

```
laatste(X,[X]).
laatste(X,[_|Y]) :- laatste(X,Y).
```

Blz. 196, it. 16.9: Hier is een recursieve definitie van de omkeerprocedure:

– De lege lijst omkeren levert de lege lijst op.
– Een niet-lege lijst kan als volgt worden omgekeerd: hak de kop eraf, keer dan de staart om, en plak tenslotte de afgehakte kop achter aan de omgekeerde staart.

In PROLOG ziet één en ander er zo uit:

```
keerom([],[]).
keerom([Kop|Staart],Lijst) :-
    keerom(Staart,Omgekeerd),
    concat(Omgekeerd,[Kop],Lijst).
```

Blz. 199, it. 16.10: De natuurlijke getallen met nul en de operaties voor optellen en aftrekken vormen geen substructuur van de gehele getallen met diezelfde operaties omdat de natuurlijke getallen niet afgesloten zijn onder de operatie *aftrekken*: als m en n natuurlijke getallen zijn met $m < n$ dan is $m - n$ geen natuurlijk getal.

Blz. 201, it. 17.1: ⟨*getal*⟩ ::= ⟨*cijfer*⟩ | ⟨*cijfer*⟩ ⟨*getal*⟩

Blz. 202, it. 17.2: 1. `x := x + 1` heeft als effect dat de waarde van de variabele x met 1 wordt verhoogd.

2. `(als x < 0 dan x := -x)` doet niets wanneer x een positief getal bevat; haalt het minteken weg wanneer x een negatief getal bevat.
3. `(als x > y dan m := x anders m := y)` vult de variabele m met het maximum van x en y.
4. `begin z := x; x := y; y := z einde` wisselt de waarden van x en y om met behulp van z; z en y bevatten na afloop het getal dat eerst in x was opgeslagen.
5. `begin x := y; y := x einde` kopieert de waarde van y in de variabele x; de oorspronkelijke inhoud van y blijft ongewijzigd; de oorspronkelijke inhoud van x gaat verloren.

6.
```
begin
  i := 0;
  p := 0;
  zolang i < x doe
    begin p := p + y; i := i + 1 einde
einde
```
berekent het product van x en y, met het resultaat in p, mits x voordat het programma begint groter dan of gelijk is aan 0. Wanneer aan deze voorwaarde niet is voldaan stopt het programma meteen (dan $p = i = 0$); in het andere geval bevat `i` na afloop de waarde x.

7.
```
begin
  i := 0;
  f := 1;
  zolang i < x doe
    begin i := i + 1; f := (i * f) einde
einde
```
berekent de faculteitsfunctie voor x, met het resultaat in `f`, mits x bij het opstarten van het programma een positieve waarde heeft. Als aan die voorwaarde is voldaan zal i na afloop van het programma de waarde x bevatten; in het andere geval stopt het programma meteen met voor `f` de waarde 1.

Blz. 202, it. 17.3: Een juiste specificatie is de volgende:

$\{x = X, y = Y\}$
```
    begin x := y; y := x einde
```
$\{x = Y, y = Y\}$.

Blz. 202, it. 17.4: De specificatie $\{\varphi\}\ \pi\ \{\top\}$ is juist voor *elke* beginconditie φ en *elk* programma π. Met andere woorden: deze specificatie zegt in feite niets. Ook zegt de specificatie *niet* dat π termineert.

Blz. 203, it. 17.5: Vervangen van de beginvoorwaarde $x \geq 0$ door $\top$ maakt dat de specificatie het volgende zegt: als het programma termineert dan zal de variabele p na afloop het product van x en y bevatten. Het programma termineert ook altijd, maar wel met de waarde $p = 0$ als x negatief is; in dat geval is $p \neq x \cdot y$ (tenzij $y = 0$).

Blz. 203, it. 17.6: Bijvoorbeeld:

$\{x > 0\}$
```
    begin
      i := 0;
      y := 0;
      zolang i < x doe
        begin i := i + 1; y := y + i einde
    einde
```
$\{y = 1 + 2 + \cdots + x\}$.

Blz. 203, it. 17.7: Het gevraagde programma is een variant op dat uit de vorige opdracht:

$\{x > 0\}$
```
    begin
      i := 0;
      y := 0;
      zolang i < x doe
        begin i := i + 1; y := y + (i*(i*i)) einde
      einde
```
$\{y = 1^3 + 2^3 + \cdots + x^3\}$.

Blz. 203, it. 1: Statement, concatenatie (achter elkaar zetten van statements), en if-then-else. Zodra een programma een while-statement bevat geldt niet meer met zekerheid dat $|T(b,\pi)| = 1$. Immers, een while statement kan aanleiding geven tot een oneindige lus.

Blz. 203, it. 2: De bewering $|T(b,\pi)| = 1$ zegt dat het aantal eind-toestanden waarin π ons kan brengen na te zijn gestart in begintoestand b, gelijk is aan 1. Dit zegt ons twee dingen. Ten eerste dat π deterministisch is (telkens hetzelfde verloopt). Immers, de eindtoestand is steeds dezelfde. Ten tweede dat π stopt.

De bewering $|T(b,\pi)| = 0$ zegt dat het aantal eind-toestanden waarin π ons kan brengen na te zijn gestart in begintoestand b, gelijk is aan 0. Dit zegt ons dat π niet stopt. Als het wel stopte moet er tenminste één eindtoestand zijn, maar hier is er geen een.

Blz. 203, it. 17.8: Volgens Definitie 17.3 geldt: $\mathcal{M} \models \{\varphi\}\,\pi\,\{\psi\}$ als en slechts als voor alle $b_2 \in T(b_1,\pi)$ geldt: als $\mathcal{M}, b_1 \models \varphi$ dan $\mathcal{M}, b_2 \models \psi$. Maar $T(b_1,\pi)$ is leeg. Dus geldt de voorgaande implicatie ledigerwijs.

Blz. 205, it. 17.9: Een juiste specificatie is:

$\{x = X\}$
```
    (als x < 0 dan x := -x)
```
$\{x = -X \vee x \geq 0\}$.

Deze specificatie kan als volgt worden afgeleid met de **'Als–dan' regel**:

$$\dfrac{\begin{array}{ll}\vdash & \{x = X \wedge x < 0\}\ \texttt{x := -x}\ \{x = -X \vee x \geq 0\} \\ \vdash & (x = X \wedge x \geq 0) \rightarrow (x = -X \vee x \geq 0)\end{array}}{\vdash \{x = X\}(\texttt{als x < 0 dan x := -x})\{x = -X \vee x \geq 0\},}$$

waarbij de specificatie in de eerste premisse volgt met behulp van het toekenningsaxioma, de regel voor versterking van de beginvoorwaarde en de regel voor afzwakking van de eindvoorwaarde. De laatste twee regels kunnen immers ook altijd gebruikt worden om voorwaarden te vervangen door logisch equivalente formules.

N.B. Een meer informatieve correctheidsbewering verkrijgen we door, bij dezelfde beginvoorwaarde, de eindvoorwaarde te versterken tot:

$$(x = X \wedge X \geq 0) \vee (x = -X \wedge X < 0).$$

De afleiding verloopt analoog aan de voorgaande.

Blz. 205, it. 17.10: Hier is de specificatie:

$\{x = X \ \wedge\ y = Y\}$
```
   (als x > y dan m := x anders m := y)
```
$\{(m = X \ \wedge\ X > Y) \ \vee\ (m = Y \ \wedge\ X \leq Y)\}$.

Deze specificatie kan als volgt worden afgeleid met de **'Als–dan–anders' regel**:

$$\begin{array}{ll} \vdash & \{x = X \wedge y = Y \wedge X > Y\}\ \texttt{m := x} \\ & \quad \{(m = X \wedge X > Y) \vee (m = Y \wedge X \leq Y)\} \\ \vdash & \{x = X \wedge y = Y \wedge X \leq Y\}\ \texttt{m := y} \\ & \quad \{(m = X \wedge X > Y) \vee (m = Y \wedge X \leq Y)\} \\ \hline \vdash & \{x = X \wedge y = Y\}\ \texttt{(als x > y dan m := x anders m := y)} \\ & \quad \{(m = X \wedge X > Y) \vee (m = Y \wedge X \leq Y)\}. \end{array}$$

De premissen in deze afleiding kunnen elk worden verkregen door een toepassing van het toekenningsaxioma, de regel voor versterking van de beginvoorwaarde en de regel voor afzwakking van de eindvoorwaarde.

Blz. 206, it. 1: Stel

$$\mathscr{M}, b_1 \models x = 3 \tag{15}$$

en

$$b_2 \in T(b_1, y := x * x). \tag{16}$$

Te bewijzen dat $\mathscr{M}, b_2 \models y = 9$. Dus te bewijzen dat $b_2(y) = 9$. Uit (15) volgt $b_1(x) = 3$. Uit (16) volgt

$$\begin{aligned} b_2 &= b_1(y|b_1(x * x)) \\ &= b_1(y|b_1(x)b_1(x)) \\ &= b_1(y|3 \times 3) \\ &= b_1(y|9). \end{aligned}$$

Hiermee $b_2(y) = b_1(y|9)(y) = 9$, wat bewezen moest worden.

Blz. 206, it. 2: Stel

$$\mathscr{M}, b_1 \models x = 3$$

en

$$b_2 \in T(b_1, y := z * z). \tag{17}$$

Te bewijzen dat $\mathscr{M}, b_2 \models y = z^2$. Dus te bewijzen dat $b_2(y) = b_2(z * z)$. We kunnen alvast verklappen dat de eerste voorwaarde er niet toe doet. Die komt dus niet meer voor in onze redenering. Uit (17) volgt $b_2 = b_1(y|b_1(z * z))$. Hiermee

$$\begin{aligned} b_2(y) &= b_1(y|b_1(z * z))(y) \\ &= b_1(z * z) \\ &= b_1(z) * b_1(z) \\ &= b_2(z) * b_2(z). \end{aligned}$$

De laatste gelijkheid volgt uit het feit dat y en z ongelijke symbolen zijn en dat b_1 en b_2 buiten y (en dus op z) samenvallen.

Blz. 206, it. 3: Stel

$$\mathscr{M}, b_1 \models x \leq 3 \tag{18}$$

en

$$b_2 \in T(b_1, \textbf{als } x \leq 4 \textbf{ dan } y := x \textbf{ anders } y := 2 * x \tag{19}$$

Te bewijzen dat $\mathscr{M}, b_2 \models xy \leq 17$. Dus te bewijzen dat $b_2(x)b_2(y) \leq 17$. Uit (18) volgt $b_1(x) \leq 3$. Omdat $\mathscr{M}, b_1 \models x \leq 4$ is (19) per definitie van de if-then-else semantiek (Def. 17.2, blz. 203) gelijkwaardig aan

$$b_2 \in T(b_1, y := x).$$

Uit dit laatste volgt $b_2 = b_1(y|b_1(x))$. Hieruit volgt dat $b_2(y) = b_1(y|b_1(x))(y) = b_1(x) \leq 3$. Dus $b_1(x)b_2(y) \leq 3 \times 3 = 9 \leq 17$. Nu volgt het gevraagde.

Blz. 206, it. 4: Stel

$$\mathscr{M}, b_1 \models x \leq 3 \tag{20}$$

en

$$b_2 \in T(b_1, \textbf{als } x \leq 1 \textbf{ dan } y := x \textbf{ anders } y := x - 1) \tag{21}$$

Te bewijzen dat $\mathscr{M}, b_2 \models xy \leq 9$. Dus te bewijzen dat $b_2(x)b_2(y) \leq 9$. Uit (20) volgt $b_1(x) \leq 3$. Deze informatie blijkt onbruikbaar. We weten namelijk hiermee niet of $\mathscr{M}, b_1 \models x \leq 1$ danwel $\mathscr{M}, b_1 \not\models x \leq 1$. We gaan dus beide takken na.

Stel allereerst $\mathscr{M}, b_1 \models x \leq 1$. Dan is volgens de if-then-else semantiek (21) gelijkwaardig aan

$$b_2 \in T(b_1, y := x),$$

wat volgens de semantiek van het assignment statement zou betekenen dat $b_2 = b_1(y|b_1(x))$. Hieruit volgt

$$\begin{aligned} b_2(x)b_2(y) &= b_1(y|b_1(x))(x) \times b_1(y|b_1(x))(y) \\ &= b_1(x)b_1(x) \\ &= 3 \times 3 \\ &= 9 \\ &\leq 9, \end{aligned}$$

hetgeen gevraagd werd in de post-conditie.

Stel nu $\mathscr{M}, b_1 \not\models x \leq 1$. Dan is volgens de if-then-else semantiek (21) gelijkwaardig aan

$$b_2 \in T(b_1, y := x - 1),$$

wat volgens de semantiek van het assignment statement zou betekenen dat $b_2 = b_1(y|b_1(x) - 1)$. Hieruit volgt

$$\begin{aligned} b_2(x)b_2(y) &= b_1(y|b_1(x) - 1)(x) \times b_1(y|b_1(x) - 1)(y) \\ &= b_1(x)(b_1(x) - 1) \\ &= 3 \times 2 \\ &= 6 \\ &\leq 9, \end{aligned}$$

hetgeen gevraagd werd.

Blz. 206, it. 1: We moet bewijzen

$$\begin{array}{ll} \vdash & \{\varphi\}\ \pi_1\ \{\psi\} \qquad (*) \\ \vdash & \{\psi\}\ \pi_2\ \{\chi\} \qquad (**) \\ \hline \vdash & \{\varphi\}\ \pi_1;\pi_2\ \{\chi\}. \end{array}$$

Stel hiervoor

$$\mathscr{M}, b_1 \models \varphi \tag{22}$$

en

$$b_3 \in T(b_1, \pi_1;\pi_2). \tag{23}$$

(De indices van de bedelingen b_i, $i = 1,3$ zijn natuurlijk niet toevallig gekozen.) Te bewijzen dat $\mathscr{M}, b_3 \models \chi$.

Uit (23) volgt dat er een b_2 is, zo dat

$$b_2 \in T(b_1, \pi_1) \tag{24}$$

en

$$b_3 \in T(b_2, \pi_2). \tag{25}$$

Uit (22), (24) en *(*)* volgt

$$\mathscr{M}, b_2 \models \psi. \tag{26}$$

Uit (26), (25) en *(**)* volgt dan

$$\mathscr{M}, b_3 \models \chi.$$

Blz. 206, it. 2: We moeten bewijzen:

$$\begin{array}{ll} \vdash & \{\varphi \wedge \sigma\}\ \pi\ \{\chi\} \qquad (*) \\ \vdash & (\varphi \wedge \neg\sigma) \to \chi \qquad (**) \\ \hline \vdash & \{\varphi\}\ (\textbf{als}\ \ \sigma\ \textbf{dan}\ \pi)\ \{\chi\}. \end{array}$$

Stel hiervoor

$$\mathscr{M}, b_1 \models \varphi \tag{27}$$

en

$$b_2 \in T(b_1,\ \textbf{als}\ \sigma\ \textbf{dan}\ \pi\ ). \tag{28}$$

Te bewijzen dat $\mathscr{M}, b_2 \models \chi$. Uit (28) volgt dat

$$\begin{cases} b_2 \in T(b_1, \pi) & \text{als } \mathscr{M}, b_1 \models \sigma \\ b_2 = b_1 & \text{anders.} \end{cases} \tag{29}$$

We onderscheiden twee gevallen: $\mathscr{M}, b_1 \models \sigma$ en $\mathscr{M}, b_1 \not\models \sigma$.

1. Stel
$$\mathscr{M}, b_1 \models \sigma. \tag{30}$$
Uit (27) en (30) volgt
$$\mathscr{M}, b_1 \models \varphi \wedge \sigma. \tag{31}$$
Uit (29.1), (31) en *(*)* volgt $\mathscr{M}, b_2 \models \chi$.

2. Stel
$$\mathscr{M}, b_1 \not\models \sigma. \tag{32}$$
Uit (27) en (32) volgt
$$\mathscr{M}, b_1 \models \varphi \wedge \neg\sigma. \tag{33}$$
Uit (29.2), (33) en *(**)* volgt nu $\mathscr{M}, b_2 \models \chi$.

Blz. 206, it. 3: We moeten bewijzen:

$$\begin{array}{ll} \vdash & \{\varphi \wedge \sigma\}\ \pi_1\ \{\psi\} \qquad (*) \\ \vdash & \{\varphi \wedge \neg\sigma\}\ \pi_2\ \{\psi\} \qquad (**) \\ \hline \vdash & \{\varphi\}\ (\textbf{als}\ \sigma\ \textbf{dan}\ \pi_1\ \textbf{anders}\ \pi_2)\ \{\psi\}. \end{array}$$

Stel hiervoor

$$\mathscr{M}, b_1 \models \varphi \tag{34}$$

en

$$b_2 \in T(b_1,\ \textbf{als}\ \sigma\ \textbf{dan}\ \pi_1\ \textbf{anders}\ \pi_2\ ). \tag{35}$$

Te bewijzen dat $\mathscr{M}, b_2 \models \psi$. Uit (35) volgt dat

$$\begin{cases} b_2 \in T(b_1, \pi_1) & \text{als } \mathscr{M}, b_1 \models \sigma \\ b_2 \in T(b_1, \pi_2) & \text{anders.} \end{cases} \tag{36}$$

We onderscheiden twee gevallen: $\mathscr{M}, b_1 \models \sigma$ en $\mathscr{M}, b_1 \not\models \sigma$.

1. Stel
$$\mathscr{M}, b_1 \models \sigma. \tag{37}$$
Uit (34) en (37) volgt
$$\mathscr{M}, b_1 \models \varphi \wedge \sigma. \tag{38}$$
Uit (36.1), (38) en *(*)* volgt $\mathscr{M}, b_2 \models \psi$.

2. Ingeval $\mathscr{M}, b_1 \not\models \sigma$ verloopt het bewijs volledig analoog aan het bewijs van geval 1.

Blz. 208, it. 1: Trek, door herhaaldelijk toepassen van de assignment-regel, de post-conditie naar boven:

$$\begin{array}{l} \{5+7=12\} \\ x := 2; \\ \{5+7=12\} \\ y := 8; \\ \{5+7=12\} \\ x := 5; \\ \{x+7=12\} \\ y := 7; \\ \{x+y=12\} \end{array}$$

Daar $(x = 2 \wedge y = 3) \to 5+7=12$ volgt de correctheid van het besproken Hoare tripel uit Regel 17.1 en herhaaldelijk toepassen van de concatenatie-regel.

Blz. 208, it. 2: De werkwijze is hetzelfde als in het vorige onderdeel.

$$\begin{array}{l} \{((x+y)-y)-y=-1\} \\ x := x+y; \\ \{(x-y)-y=-1\} \\ x := x-y; \\ \{x-y=-1\} \\ y := x-y; \\ \{y=-1\} \end{array}$$

Daar $((x+y)-y)-y=-1$ geïmpliceerd wordt door $x = 2 \wedge y = 3$, volgt het gestelde, ook weer uit Regel 17.1 en herhaaldelijk toepassen van de concatenatie-regel.

Blz. 208, it. 3: Trek, door herhaaldelijk toepassen van de assignment-regel, de post-conditie naar boven:

$\{((x+y)-y)(((x+y)-y)-y) = -2\}$
$x := x+y;$
$\{(x-y)((x-y)-y) = -2\}$
$x := x-y;$
$\{x(x-y) = -2\}$
$y := x-y;$
$\{xy = -2\}$

De bewering $((x+y)-y)(((x+y)-y)-y) = -2$ kan vereenvoudigd worden tot $2+x^2 = xy$. Deze bewering wordt door $x = 2 \wedge y = 3$ geïmpliceerd.

Blz. 208, it. 4: De naar boven gewerkte post-conditie ziet er uit als

$$\{-2 * (x-w) = 2\}.$$

Deze wordt inderdaad geïmpliceerd door $x = 2 \wedge w = 3$.

Blz. 208, it. 5: De omhoog gewerkte post-conditie ziet er uit als

$$\{-2 * (x-w) \leq 3\}$$

Deze bewering wordt geïmpliceerd door $x = 2 \wedge w = 3$, want inderdaad geldt $2 \leq 3$.

Blz. 209, it. 1:
$(x = 2 \rightarrow y = 3) \wedge (x \neq 2 \rightarrow x = 3)$
als $x = 2$
dan
$\{y = 3\}$
$x := y$
$\{x = 3\}$
anders
$\{x = 3\}$
$y := x$
$\{x = 3\}$
;
$\{x = 3\}$

Blz. 212, it. 1: Bij een while loop gaan we altijd eerst zoek naar een invariant η. Maak een tabelletje. Omdat de waarde van x onbekend is vullen voor x maar wat in. (Niet te hoog, anders neemt de executie te veel stappen in beslag.) Neem $x = 3$.

a	x	y
3	3	0
2	3	1
1	3	2
0	3	3

Als $a = 0$ worden het programma uit de while loop gegooid. Wat is een goede invariant (zonder op de getallen zelf te letten)? Na enig kijken lijkt $\eta = (a+y = x)$ wel een goede. Dat klopt, want η wordt geïmpliceerd door de conditie vlak voor de while loop, en impliceert samen met de negatie van de test $a = 0$ de postconditie $x = y$:

$\{x \geq 0\}$
$\{x+0 = x\}$
$a := x;$
$\{a+0 = x\}$
$y := 0;$
$\{a+y = x\}$
zolang $a <> 0$ **doe** {
$y := y+1;$
$a := a-1;$
}
$\{a+y = x \wedge a = 0\}$
$\{x = y\}$

Nu nog het binnenste deel van de while loop correct bewijzen. Volgens de while regel is het daarvoor voldoende om $\eta \wedge \sigma = (a+y = x \wedge a \neq 0)$ als pre-conditie, en $\eta = (a+y = x)$ als post-conditie te nemen:

$\{x \geq 0\}$
$\{x+0 = x\}$
$a := x;$
$\{a+0 = x\}$
$y := 0;$
$\{a+y = x\}$
zolang $a <> 0$ **doe** {
$\{a+y = x \wedge a \neq 0\}$ (*)
$\{a-1+(y+1) = x\}$ (**)
$y := y+1;$
$\{(a-1)+y = x\}$
$a := a-1;$
$\{a+y = x\}$
}
$\{a+y = x \wedge a = 0\}$
$\{x = y\}$

Daar *(**)* wordt geïmpliceerd door *(*)*, is de afleiding rond. Het bewijs ziet er dan zo uit:

1.	$\{x \geq 0\}$	pre-conditie
2.	$\{x+0 = x\}$	assignment-regel 3,4
3.	$a := x;$	
4.	$\{a+0 = x\}$	assignment-regel 5,6
5.	$y := 0;$	
6.	$\{a+y = x\}$	Zolang-regel, 8-13
7.	**zolang** $a <> 0$ **doe** {	
8.	$\{a+y = x \wedge a \neq 0\}$	invariant en guard
9.	$\{a-1+(y+1) = x\}$	assignment-regel 10,11
10.	$y := y+1;$	
11.	$\{(a-1)+y = x\}$	assignment-regel 12,13
12.	$a := a-1;$	
13.	$\{a+y = x\}$	invariant
14.	}	
15.	$\{a+y = x \wedge a = 0\}$	Zolang-regel, 8-13
16.	$\{x = y\}$	post-conditie

Blz. 212, it. 2: Executietabel binnen while loop voor willekeurige x:

x	y
3	0
3	1
3	2
3	3

Invariant: $\eta = (y \leq x)$.

Blz. 212, it. 3: Executietabel binnen while loop voor willekeurige x en y:

a	x	y	z
0	2	3	0
1	2	3	2
2	2	3	4
3	2	3	6

Invariant: $\eta = (ax = z)$.

Blz. 212, it. 4: Executietabel binnen while loop voor willekeurige x en y:

x	y	y_0	z
2	3	3	0
2	2	3	2
2	1	3	4
2	0	3	6

De variabele y houdt bij hoeveel keer x nog bij z opgeteld moet worden. De expressie $y_0 - y$ geeft dus de factor in opbouw in het onaffe product $x(y_0 - y)$. Een geschikte invariant is dus $\eta = x(y_0 - y)$.

Blz. 212, it. 17.16: Hint: maak een tabelletje waar het verloop van alle variabelen in de while loop kan worden gevolgd. Dit suggereert een invariant.

Blz. 214, it. 17.17: Kijk voor de executietabellen in het antwoord van Opgave 17.15 op blz. 212. Met behulp van deze executietabellen kunnen geschikte varianten worden opgesteld.

– **Copy1** (onderdeel 1 op blz. 212): variant $t = a$.

  Immers,

$$\sigma = (a \neq 0),$$

$$\pi \;=\; y := y + 1;\, a := a - 1,$$

  zodat

$$\{\sigma, t \geq 0, t = x\} \;\; \pi \;\; \{t \geq 0, t < x\}$$

  correct is:

$$\begin{array}{l} \{a \neq 0 \wedge a \geq 0 \wedge a = x\} \\ \{a - 1 \geq 0 \wedge a - 1 < x\} \\ y := y + 1; \\ \{a - 1 \geq 0 \wedge a - 1 < x\} \\ a := a - 1 \\ \{a \geq 0 \wedge a < x\} \end{array}$$

  Immers,

$$\begin{array}{l} (a \neq 0 \wedge a \geq 0 \wedge a = x) \\ \qquad \leftrightarrow (a > 0 \wedge a = x) \rightarrow (a - 1 \geq 0 \wedge a - 1 < x). \end{array}$$

  Nu kan het bewijs worden afgemaakt.

– **Copy2** (onderdeel 2 op blz. 212): variant $t = x - y$.

– **Multi1** (onderdeel 3 op blz. 212): variant $t = y - a$.

– **Multi2** (onderdeel 4 op blz. 212): variant $t = y$.

Andere varianten kunnen ook goed zijn.

Blz. 216, it. 17.19: Laten we kijken wat er gebeurt als q op zichzelf wordt losgelaten, i.e., we gaan kijken wat er gebeurt bij de uitvoering van $q(q)$.

– Volgens q's *implementatie* geldt dan dat $q(q)$ stopt als en slechts als $h(q,q) = 0$. (Ga na.)

– Anderzijds geeft $h(q,q)$ *per definitie* van h de uitkomst 0 terug als q **niet** stopt op input q.

Dat is eigenaardig: we hebben afgeleid dat $q(q)$ stopt als en slechts als $q(q)$ niet stopt. Dat kan logisch gezien niet, en dus kunnen we niet anders dan concluderen dat onze oorspronkelijke aanname niet waar kan zijn. Er kan dus geen controle-programma $h/2$ bestaan dat van een willekeurig programma $j/1$ en een willekeurige input i kan controleren of $j(i)$ stopt.

Blz. 217, it. 17.20: Probleem *i)* is beslisbaar. Dat is makkelijk in te zien: laat j een Java-programma zijn waarvan moet worden bepaald of het in 10,000 of minder instructies stopt. Simuleer j en houd tijdens de simulatie het aantal instructies bij. Rapporteer JA als de simulatie in 10,000 of minder instructies stopt. Rapporteer anders NEE en stop.

Probleem *ii)* is onbeslisbaar. Dit laten we zien door het Java/0-stop probleem (Voorbeeld 17.18, blz. 216) te reduceren naar dit probleem.

Laat j een willekeurig Java/0-programma zijn waarvan moet worden bepaald of het stopt. Laat Java-programma j' ontstaan uit j door er een laatste instructie aan toe te voegen die 10,001 of meer karakters afdrukt. (Bijvoorbeeld 10,001 spaties. Om er voor te zorgen dat j' niet eerst vanwege j al vroegtijdig karakters gaat afdrukken, laten we j' alle print-statements van j "inslikken," bijvoorbeeld door ze naar een ander bestand te leiden.) Eenvoudig is na te gaan dat j stopt als en slechts als j' 10,000 of meer karakters afdrukt. Als probleem *ii)*) beslisbaar zou zijn, zouden we via j' nu ook kunnen beslissen of j stopt. Dat is in tegenspraak met de onbeslisbaarheid van het stop-probleem, dus probleem (*ii)* is onbeslisbaar.

Blz. 217, it. 1: We reduceren het Java/0-stop probleem (Voorbeeld'17.18, blz. stop-probleem-0) naar dit probleem. Laat j een willekeurig Java/0-programma zijn waarvan moet worden bepaald of het stopt. Om interferentie te voorkomen make we j a-vrij. We voeren een globale substitutie uit op j, zó dat evt. voorkomens van $a \in A$ vervangen worden door een volledig nieuwe teken $a \notin A$. We krijgen een a-=vrij programma j'. Bekijk nu programma g:

g: voer j' uit en druk a af als j' stopt.

Eenvoudig is na te gaan dat g het symbool a afdrukt als en slechts als j' stopt. Omdat we hebben aangenomen dat het print-probleem beslisbaar is, kunnen we beslissen of g ooit a afdrukt, en dus of j' stopt. Dat laatste is in tegenspraak met de onbeslisbaarheid van het inputloze stop-probleem.

Blz. 217, it. 2: We reduceren het Java/0-stop probleem naar dit probleem. Laat j/0 een willekeurig Java/0-programma zijn waarvan moet worden bepaald of het stopt. We laten het test-programma het programma j/0 omzetten naar een programma j'/1 dat *niets* met de input doet. Eenvoudig is na te gaan dat j/0 stopt als en slechts als j'/1 op alle inputs stopt. (Immers inhoud van de input wordt toch niet gebruikt.)

Blz. 217, it. 3: We reduceren het Java/0-stop probleem naar dit probleem. Laat j/0 een willekeurig Java/0-programma zijn waarvan moet worden bepaald of het stopt. Laat $s \in J$ een triviaal (en liefst zo kort mogelijk) programma zijn dat niets afdrukt en altijd stopt. Het test-programma g verandert j in j' door j' alle print-statements van j te laten "inslikken," bijvoorbeeld door ze naar een ander bestand (of `/dev/null/`) te leiden. Vervolgens gebruikt g het algoritme dat test of twee programma's zich hetzelfde gedragen om te testen of j en s zich hetzelfde gedragen. Eenvoudig is na te gaan dat g concludeert tot programma-equivalentie als en slechts als j' stopt als en slechts als j stopt. Dat zou betekenen dat g kan worden gebruikt om het stop-probleem op te lossen.

Blz. 217, it. 2: Een onbegrensde externe toestand (denk aan pagefile) p zorgt er voor dat de toestandruimte oneindig groot wordt. Nu gaat het cykel-argument niet meer op.

Blz. 221, it. 18.1: Noem de machinetoestand voordat π wordt uitgevoerd b. Wanneer φ niet waar is in b zou elke mogelijke toestand t bereikbaar zijn volgens de alternatieve definitie van T_π. Dit zou betekenen dat de modellen die meer dan 2 toestanden bevatten *indeterministisch* zouden worden. Erger, allerlei ongewenste toestanden zouden toegankelijk worden, bijvoorbeeld kunnen waarden van variabelen die niet in φ of α voorkomen veranderd zijn.

Blz. 221, it. 18.2: **als–dan–anders** Laat $\pi = (\textbf{als } \varphi \textbf{ dan } \alpha \textbf{ anders } \beta)$, en stel T_α en T_β zijn deterministisch. Kies een willekeurige toestand b. Als $b \models \varphi$ dan is er hoogstens één e zodat $bT_\pi e$ wanneer $\langle b,e\rangle \in T_\alpha$. Als $b \not\models \varphi$ dan evenzo hoogstens één e zodat $bT_\pi e$ namelijk wanneer $\langle b,e\rangle \in T_\beta$.

zolang Laat $\pi = \textbf{zolang } \varphi \textbf{ doe } \alpha$, en stel T_α is deterministisch. Kies weer een b. We onderscheiden de volgende mogelijkheden:

1. $b \not\models \varphi$. Dan is er een unieke toestand bereikbaar volgens T_π, want dan $b = e$.
2. $b \models \varphi$. Stel dat er een volgens T_π bereikbare e bestaat. Dan is die e ook uniek. Want er moet een reeks (ook unieke) tussentoestanden $b_1,\ldots b_n$ zijn geweest die φ waar maakten en die opvolgend binnen T_α bereikbaar waren, en waarna er een overgang is naar e waarop φ onwaar is. De lengte van die reeks ligt dan ook vast: voor iedere kortere reeks zou in de eindtoestand φ waar zijn, en dat is uitgesloten; langer kan ook niet want dan zou φ onwaar worden in de (dan) tussentoestand e.

Blz. 222, it. 18.3: We geven een informeel bewijs. De vierde bewering is waar op de verzameling gehele getallen als voor elke begintoestand b:

$$\forall x \forall y[\texttt{zolang niet x = y doe x := x + 1}](x = y)$$

waar is, dus wanneer voor elke b geldt dat

$$[\texttt{zolang niet x = y doe x := x + 1}](x = y)$$

en dat wil weer zeggen dat in een toestand e die uit b bereikbaar is volgens T_π (waarbij $\pi = \texttt{zolang niet x = y doe x := x + 1}$) de formule $x = y$ moet gelden. Om dit na te gaan onderscheiden we drie mogelijkheden:

1. $b(x) < b(y)$. De **zolang**-opdracht verhoogt nu de waarde van x steeds met 1, totdat x en y even groot zijn, te weten $b(y)$. In die toestand geldt uiteraard $x = y$.
2. $b(x) = b(y)$. Omdat de conditie van de **zolang**-opdracht nu niet geldt, zal nu $e = b$ het geval zijn; dus geldt in e dan $x = y$.
3. $b(x) > b(y)$. We redeneren nu analoog aan de derde bewering in het tekstvoorbeeld: door het ophogen van x zal de conditie nu altijd blijven gelden, en het programma blijft in de zolang-lus. Dynamisch volgt dit uit feit dat er dan geen bereikbare eindtoestand bestaat.

Blz. 222, it. 18.4: Als substitutie wel zou zijn toegestaan, dan resulteert dit in:

$$[x/y][\mathrm{x} := \mathrm{y} + 1](x \neq y) = [\mathrm{x} := \mathrm{x} + 1](x \neq x).$$

De laatste formule is onvervulbaar want als er een vervullende bedeling b zou bestaan, dan zou $b(x) + 1 \neq b(x) + 1$, hetgeen onmogelijk is. De premisse $\forall y[\mathrm{x} := \mathrm{y} + 1](x \neq y)$ is echter waar op de structuur $\mathbb{N}$.

Blz. 223, it. 18.5: **Bewijs:** *Opeenvolging* is semantisch geldig omdat de deelformules voor en na $\leftrightarrow$ in dezelfde toestanden waar zijn. We gaan dit na.

Stel $[\textbf{begin } \pi_1;\cdots;\pi_n \textbf{ einde}]\varphi$ is onwaar in b. Dat is precies zo als er een e bestaat zodat $\langle b,e\rangle \in T_{\textbf{begin } \pi_1;\cdots;\pi_n \textbf{ einde}}$ en φ onwaar is in e. Dit betekent dat er $e_0, e_1, \ldots, e_n$ bestaan waarvoor $e_0 = b$, $e_n = e$ en $\langle e_{i-1}, e_i\rangle \in T_{\pi_i}$ voor elke $i = 1,2,\ldots,n$ en φ onwaar is. Dit is equivalent (inductief te bewijzen) met het bestaan van $e_0(= b),\ldots,e_n$ zodat $[\pi_{k+1}]\cdots[\pi_n]\varphi$ onwaar is voor elke $k = 0,\ldots,n$; kortom als $[\pi_1]\cdots[\pi_n]\varphi$ onwaar is in b. □

Blz. 223, it. 18.6: Herformuleren houdt hier natuurlijk in: bewijzen dat beide formules equivalent zijn met gebruik van de overige principes. Vanwege *opeenvolging* geldt

$$[\alpha][\ \textbf{zolang } \psi \textbf{ doe } \alpha]\varphi \iff [\textbf{begin } \alpha;\ \textbf{zolang } \psi \textbf{ doe } \alpha \textbf{ einde}]\varphi$$

en daardoor is de rechterkant van het *iteratie-axioma* equivalent met

$$(\psi \rightarrow [\textbf{begin } \alpha; \textbf{ zolang } \psi \textbf{ doe } \alpha \textbf{ einde}]\varphi) \wedge (\neg\psi \rightarrow \varphi)$$

en met het *als–dan-axioma* reduceert dit tot

$$[(\textbf{als } \psi \textbf{ dan begin } \alpha; \textbf{ zolang } \psi \textbf{ doe } \alpha \textbf{ einde})]\varphi$$

□

Blz. 223, it. 18.7: In $[v := t]\varphi$ zal een vrij in φ voorkomende v gebonden worden, terwijl die in $[t/v]\varphi$ vervangen zal worden. Het effect is in feite hetzelfde en daarom is er geen sprake van een tegenstrijdigheid. Vergelijk de situatie bij $\exists x\, x = a$ en $a = a$. Overigens kan $[t/v]\varphi$ best nog v als vrije variabele bevatten, bij voorbeeld als $t = v$; in dat geval is t echter ook vrij in $[v := t]\varphi$. Tenslotte wijzen we erop dat open en gesloten formules best logisch equivalent kunnen zijn: denk aan $\forall x\, x = x$ en $x = x$.

Blz. 224, it. 18.8:

1	$\varphi \rightarrow [\pi_1]\psi$	[gegeven stelling]
2	$\psi \rightarrow [\pi_2]\chi$	[gegeven stelling]
3	$[\pi_1]\psi \rightarrow [\pi_1][\pi_2]\chi$	[uit 2 met stelling 18.1]
4	$[\pi_1][\pi_2]\chi \leftrightarrow [\textbf{begin } \pi_1;\pi_2 \textbf{ einde}]\chi$	[opeenvolging]
5	$[\pi_1]\psi \rightarrow [\textbf{begin } \pi_1;\pi_2 \textbf{ einde}]\chi$	[uit 3 en 4 met pL]
6	$\varphi \rightarrow [\textbf{begin } \pi_1;\pi_2 \textbf{ einde}]\chi$	[uit 1 en 5 met pL]

□

Bibliografie

[Van Amstel 1984] J.J. van Amstel, *Programmeren: het ontwerpen van algoritmen met Pascal*, Academic Service, Den Haag.

[Apt 1988] K.R. Apt, "Introduction to Logic Programming", Rapport, Centrum voor Wiskunde en Informatica, Amsterdam. Verschijnt in: J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, North Holland, Amsterdam.

[Barbeau 2000] E.J. Barbeau, *Mathematical Fallacies, Flaws, and Flimflam*, MAA Press, Washington D.C.

[Barwise 1987] J. Barwise, 'Noun Phrases, Generalized Quantifiers and Anaphora', in: P. Gärdenfors (ed.), *Generalized Quantifiers: Linguistic and Logical Approaches*, Reidel, Dordrecht.

[Barwise & Cooper 1981] J. Barwise & R. Cooper, 'Generalized Quantifiers and Natural Language', *Linguistics and Philosophy* 4, 159–219.

[Van Benthem 1986] J. van Benthem 1986, *Essays in Logical Semantics*, Reidel, Dordrecht.

[Van Benthem & Ter Meulen 1985] J. van Benthem & A. ter Meulen (eds.), *Generalized Quantifiers*, Foris, Dordrecht.

[Bradko 1986] I. Bradko, *Prolog Programming for Artificial Intelligence*, International Computer Science Series, Addison-Wesley, Wokingham, England etc.

[Bunch 1997] B. Bunch, *Mathematical fallacies and paradoxes*, Dover publications.

[Burris 1998] S.N. Burris, *Logic for mathematics and Computer Science*, Prentice Hall.

[Clocksin & Mellish 1981] W.F. Clocksin & C.S. Mellish, *Programming in Prolog*, Springer, Berlin etc.

[Van Dalen 1978] D. van Dalen, *Filosofische grondslagen van de wiskunde*, Van Gorcum, Assen.

[Van Dalen 1983] D. van Dalen, *Logic and Structure*, Springer Verlag, Berlin etc., 1983 (tweede druk).

[Van Dalen e.a. 1975] D. van Dalen, H.C. Doets, H.C.M. de Swart, *Verzamelingen; naïef, axiomatisch, toegepast*, Oosthoek, Scheltema & Holkema, Utrecht.

[Dowty 1982] D. Dowty, 'Grammatical Relations and Montague Grammar', in: P. Jacobson & G.K. Pullum (eds.), *The Nature of Syntactic Representation*, Reidel, Dordrecht.

[Doxiadis em at al. 2010] A. Doxiadis, C.H. Papadimitriou, aA. Papadatos, A.D. Donna, en P. Eusebio, *Logicomix: een epische zoektocht naar de waarheid*, Guanda Graphic Press.

[Dudley 1992] D. Underwood Dudley (1992), *Mathematical Cranks*, MAA Press, Washington D.C.

[Dummett 1973] M. Dummett, *Frege, Philosophy of Language*, Duckworth, London.

[Van Eijck 1982] J. van Eijck, *Filosofie: een inleiding*, Boom, Meppel.

[Van Eijck 1987] J. van Eijck, *Programmeren in Turbo Pascal*, Academic Service, Schoonhoven.

[Emmet 1969] E.R. Emmet, *Logisch denken*, Het Spectrum, Utrecht & Antwerpen (Aula 73).

[Enderton 1972] H.B. Enderton, *A Mathematical Introduction to Logic*, Academic Press, New York etc.

[Fitch 1952] F.B. Fitch, *ymbolic Logic, An Introduction*, The Ronald Press Company.

[Frege 1879] G. Frege, *Begriffsschrift*, Nebert, Halle.

[Gamut 1982] L.T.F. Gamut, *Logica, taal en betekenis*, Het Spectrum, Utrecht & Antwerpen (2 delen).

[Gordon 1988] M.J.C. Gordon, *Programming Language Theory and its Implementation*, Prentice Hall, New York etc.

[Gries 1983] D. Gries, *The Science of Programming*, Springer Verlag, Berlin etc. (second edition).

[Groenendijk & Stokhof 1987] J. Groenendijk & M. Stokhof, 'Dynamic Predicate Logic: towards a Compositional and Non-Representational Discourse Theory', ongepubliceerd manuscript, ITLI, Universiteit van Amsterdam.

[Halmos 1960] P. Halmos, *Naive set theory*. Princeton, NJ: D. Van Nostrand Company, 1960. Herdukt door Springer-Verlag, New York, 1974.

[Harel 1979] D. Harel, *First-Order Dynamic Logic*, Lecture Notes in Computer Science, Volume 68, Springer-Verlag, Berlijn.

[Harel 1984] D. Harel, 'Dynamic Logic', in: D. Gabbay & F. Günthner (eds.), *Handbook of Philosophical Logic, Volume 2*, Reidel, Dordrecht.

[Heim 1982] I. Heim, *The Semantics of Definite and Indefinite Noun Phrases*, dissertatie, University of Massachusetts, Amherst.

[Hoek, van der 1995] W. van der Hoek, J.-W. Klop, J.-J. Meyer, R.C. de Vrijer, *Formele Methoden*, Dictaat, Universiteit Utrecht..

[Hofstadter 1979] D.R. Hofstadter, *Gödel, Escher, Bach: an Eternal Golden Braid*, Basic Books, New York.

[Hughes & Becht 1978] P. Hughes & G. Becht, *Vicious Circles and Infinity; an Anthology of Paradoxes*, Penguin Books, Harmondsworth.

[Hughes & Cresswell 1968] G.E. Hughes & M.J. Cresswell, *An Introduction to Modal Logic*, Methuen, London.

[Humphreys 1951] C. Humphreys, *Buddhism*, Penguin, Harmondsworth.

[Jeffrey 1967] R.C. Jeffrey, *Formal logic. Its Scope and Limits*, McGraw-Hill, New-York.

[Jensen & Wirth 1974] K. Jensen & N. Wirth, *Pascal User Manual and Report*, Springer Verlag, New York etc. (Third edition: Revised for the ISO Pascal Standard, 1985).

[Kamp 1981] H. Kamp, 'A Theory of Truth and Semantic Representation'. in: Groenendijk, Janssen & Stokhof (eds.), *Formal Methods in the Study of Language*, volume 1, Mathematical Centre Tracts 135, Amsterdam; herdrukt in Groenendijk, Janssen & Stokhof (eds.), *Truth, Interpretation and Information*, GRASS 2, Foris, Dordrecht 1984.

[Kant 1787] I. Kant, *Kritik der reinen Vernunft*, tweede verbeterde druk, Hartknoch, Riga.

[Lemmens *et al.* 1999] H. Zantema en P.W.H. Lemmens , *Beschrijven en Bewijzen*, Delft University Press. (Nog te vinden op het web.)

[Lewis 1972] D. Lewis, 'General Semantics', in [Davidson & Harman 1972].

[Manna *et al.* 1993] Z. Manna en R.J. Waldinger, *The deductive foundations of computer programming—a one-volume version of "The logical basis for computer programming"*, Addison-Wesley, 1993.

[Montague 1974] *Formal Philosophy, Selected Papers of Richard Montague*, edited by R.H. Thomason, Yale University Press, New Haven & London.

[Nagel & Newman 1975] E. Nagel & J. Newman, *De stelling van Gödel*, Aula, het Spectrum, Utrecht.

[Pereira & Shieber 1987] F.C.N. Pereira & S.M. Shieber, *Prolog and Natural Language Analysis*, CSLI Lecture Notes Number 10, Stanford.

[Rucker 1984] R. Rucker, *Infinity and the Mind*, Paladin Books, London.

[Russell 1905] B. Russell, 'On Denoting', in: *Mind* 14, 479–493.

[Shen & Vereshchagin 2002] A. Shen & N.K. Vereshchagin, *Basic set theory*, AMS Press.

[Smullyan 1978] R. Smullyan, *What is the Name of this Book?*, Prentice Hall, Englewood Cliffs, New Jersey.

[Smullyan 1982] R. Smullyan, *The Lady or the Tiger?*, Knopf, New York (Nederlandse vertaling: *De prinses of de tijger*, Ambo boeken, Baarn 1983).

[Steels 1983] L. Steels, *Programmeren in LISP*, Academic Service, Den Haag.

[Strawson 1950] P.F. Strawson, 'On Referring', in: *Mind* 59.

[Thijsse 1983] E. Thijsse, 'On some proposed universals of natural language', in: A. ter Meulen (ed.), *Studies in Modeltheoretic Semantics*, Foris, Dordrecht.

[De Vries 1984] G. de Vries, *De ontwikkeling van wetenschap*, Wolters-Noordhoff.

[Winfield 1983] Alan Winfield, *The Complete Forth*, Sigma Technical Press, Wilmslow, Cheshire, UK (in het Nederlands verschenen bij Academic Service, Den Haag, onder de titel *Flitsend Forth*).

[Wirth 1976] N. Wirth, *Algorithms + Data Structures = Programs*, Prentice Hall, Englewood Cliffs, New Jersey.

[Zwarts 1981] F. Zwarts, 'Negatief polaire uitdrukkingen', in: *Glot* 4.

[Zwarts 1986] F. Zwarts, *Categoriale Grammatica en Algebraïsche Semantiek*, dissertatie, Groningen.

Index