Logic for Computer Science

13 – Natural deduction

Wouter Swierstra

University of Utrecht

Last time

Games

This lecture

Natural deduction

Recap

So far, we have encountered propositional logic in several lectures:

- The first lecture defined the syntax of propositional logic informally.
- Later, we saw how to define this syntax formally as an inductively defined set.
- We have studied the semantics of propositional logic using truth tables.
- We have seen the semantics of propositional logic informally using proof strategies.

Can we not give a more precise definition of proof?

And relate it to the 'truth table semantics' we saw in the first lecture?

What is a proof?

Given a formula in propositional logic *p*, we can check when *p* holds for all possible values of its atomic propositional variables – this is what we do when we write a truth table.

We can also give a 'proof sketch' using proof strategies – but we haven't made precise what these strategies are, relying on an informal diagrammatic description.

Can we define a set of all proofs of some propositional logic formula?

After all, we managed to define the syntax of propositionial logic as inductively defined set – can we do the same for its semantics?

Doing so would provide a definitive answer to the question *"what is a proof?"* (in propositional logic).

Syntax and semantics

We can define the *syntax* of propositional logic using BNF as follows:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

Can we define a *semantics*, describing the set of valid proofs for an arbitrary propositional formula?

Inductively defined relations

So far, we have seen the BNF notation for inductively defined sets.

But what notation should we use for inductively defined *relations*?

For example, we defined the $\leqslant$ *relation* between Peano natural numbers using the following rules:

- for all $n \in \mathbb{N},\ 0 \leqslant n$;
- if $n \leqslant m$, then $s(n) \leqslant s(m)$

Isn't there a better notation?

Notation for inductively defined relations

Inductively defined relations are often given by means of *inference rules*:

$$\frac{}{0 \leqslant n} \leqslant\text{-Base}$$

$$\frac{n \leqslant m}{s(n) \leqslant s(m)} \leqslant\text{-Step}$$

Here we have two inference rules, named Base and Step; these rules together define a relation $(\leqslant) \ \subseteq \ \mathbb{N} \times \mathbb{N}$.

The statements above the horizontal line are the *premises* - the assumptions that you must establish in order to use this rule; the statement under the horizontal line is the *conclusion* that you can draw from these assumptions.

Notation for inductively defined relations

These rules state that there are two ways to prove that $n \leqslant m$:

$$\frac{}{0 \leqslant n} \leqslant\text{-Base}$$

$$\frac{n \leqslant m}{s(n) \leqslant s(m)} \leqslant\text{-Step}$$

- if $n = 0$ the $\leqslant$-Base rule tells us that $0 \leqslant n$ – for *any* n;
- if we can show $n \leqslant m$, we can use the $\leqslant$-Step rule to prove $s(n) \leqslant s(m)$.

A rule without premises is called an *axiom*.

Writing proofs

By repeatedly applying these rules, we can write larger proofs.

For example, to give a formal proof that $2 \leqslant 5$ we write:

$$\dfrac{\dfrac{\dfrac{}{0 \leqslant s(s(s(0)))}\ \leqslant\text{-Base}}{s(0) \leqslant s(s(s(s(0))))}\ \leqslant\text{-Step}}{s(s(0)) \leqslant s(s(s(s(s(0)))))}\ \leqslant\text{-Step}$$

We can read these rules top-to-bottom or bottom-to-top.

Such a proof is sometimes referred to a as *derivation*.

Each of the inference rules gives a different 'lego piece' that we can use to write bigger proofs.

Example: even numbers

We can use these inference rule notation to write all kinds of relations - not just less-than-or-equals.

For example, we may want to define the unary relation isEven – that proves that a given number is even.

$$\frac{}{\text{isEven}(0)}\ \text{isEven-Base}$$

$$\frac{\text{isEven}(n)}{\text{isEven}(s(s(n)))}\ \text{isEven-Step}$$

Question

Give a derivation that s(s(s(s(0)))) is even.

Example: isSorted

Similarly, we can define inference rules that make precise when a list of numbers is sorted:

$$\frac{}{\text{isSorted}([\,])}\ \text{isSorted-empty}$$

$$\frac{}{\text{isSorted}(n : [\,])}\ \text{isSorted-Single}$$

$$\frac{n \leqslant m \qquad \text{isSorted}(m : w)}{\text{isSorted}(n : m : w)}\ \text{isSorted-Step}$$

Note that we can require more than one hypothesis – as in the isSorted-Step rule (don't worry about the proving $n \leqslant m$ in the step-rule).

Question

Prove that the list $1 : 3 : 5 : [\,]$ is indeed sorted.

Exercise

A word over an alphabet Σ is called a **palindrome** if it reads the same backward as forward.

Examples include: ‘racecar’, ‘radar’, or ‘madam’.

Question

Give a inference rules that characterise a unary relation on words, capturing the fact that they are a palindrome.

Exercise

A word over an alphabet Σ is called a **palindrome** if it reads the same backward as forward.

Examples include: 'racecar', 'radar', or 'madam'.

Question

Give a inference rules that characterise a unary relation on words, capturing the fact that they are a palindrome.

$$\frac{}{\text{isPalindrome}(\varepsilon)}\ \text{isPalindrome-empty}$$

$$\frac{a \in \Sigma}{\text{isPalindrome}(a)}\ \text{isPalindrome-Single}$$

$$\frac{a \in \Sigma \qquad \text{isPalindrome}(w)}{\text{isPalindrome}(a\,w\,a)}\ \text{isPalindrome-Step}$$

Challenge

Given the following set of propositional logical formulas over a set of atomic variables P:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

Can we give inference rules that capture precisely the tautologies?

Challenge

Given the following set of propositional logical formulas over a set of atomic variables P:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

Can we give inference rules that capture precisely the tautologies?

Yes!

These inference rules, sometimes called *natural deduction*, formalize the proof strategies that we have seen previously.

Natural deduction

Most logical textbooks do not introduce an explicit name for the relation capturing 'truthfulness' of a given propositional logical formula, writing:

$$\frac{P \qquad Q}{P \wedge Q} \wedge\text{-I}$$

Rather than the more explicit:

$$\frac{\text{isTrue}(P) \qquad \text{isTrue}(Q)}{\text{isTrue}(P \wedge Q)} \wedge\text{-I}$$

Proof strategies vs natural deduction

Compare the proof strategy for conjunction introduction:

Proof of P

Proof of Q

Therefore we conclude $P \wedge Q$.

And the inference rule for conjunction introduction:

$$\frac{P \qquad Q}{P \wedge Q} \wedge\text{-I}$$

Conjuction elimination

$\vdots$

Proof of $P \wedge Q$

$\vdots$

Therefore, P holds.

Question

What is the corresponding elimination rule for conjunction?

Conjuction elimination

$\vdots$

Proof of $P \wedge Q$

$\vdots$

Therefore, P holds.

Question

What is the corresponding elimination rule for conjunction?

$$\frac{P \wedge Q}{P} \wedge\text{-E}_1$$

Assumptions...

Assume P.

$\vdots$

Proof of Q.

$\vdots$

Therefore, we can conclude $P \Rightarrow Q$ $\square$

But what about the implication introduction rule?

In this rule we can use the assumption P in the 'inner' box – but nowhere else. How can we account for such assumptions in our proof rules?

Assumptions and contexts

Rather than define our proof rules as a *unary* relation, we instead define them as a *binary* relation between propositional logic formulas and *the list of assumptions* we are allowed to use.

We will typically use capital greek letters, such as Γ and Δ, to refer to such lists of assumptions, also known as *contexts*. The contexts are defined inductively as follows:

$$\Gamma \quad ::= \quad \varepsilon \mid \Gamma, p$$

We will write $\Gamma \vdash P$ if we can prove the proposition logic formula P from the list of assumptions Γ.

If a formula P is provable without making any assumptions, we simply write $\vdash P$.

Conjunction rules

We can rephrase our previous rules for conjunction as follows:

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P \land Q} \land I$$

$$\frac{\Gamma \vdash P \land Q}{\Gamma \vdash P} \land E_1$$

$$\frac{\Gamma \vdash P \land Q}{\Gamma \vdash Q} \land E_2$$

These rules did not use or change the context Γ, so the rules remain largely unchanged.

Implication

The implication introduction rule, however, does add new assumptions:

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q}\ \mathsf{I}\Rightarrow$$

Here we can see how Γ may change during a derivation. To show $P \Rightarrow Q$, we add P to our list of assumptions and establish that Q holds.

Using assumptions

We need to add one last rule, explaining how to use an assumption P:

$$\frac{P \in \Gamma}{\Gamma \vdash P}\ \text{Assumption}$$

In other words, once we assume P (as it is in the context Γ), we can prove that P holds.

For example, we can now give a complete derivation showing that $\vdash P \Rightarrow P$

$$\frac{\dfrac{P \in P}{P \vdash P}\ \text{Assumption}}{\vdash P \Rightarrow P}\ \text{I}{\Rightarrow}$$

It's customary to leave out the 'trivial' checks, such as P $\in$ P, from the leaves of a derivation as they can be inferred easily enough.

Example derivation

Combining the rules we have seen so far, we can prove that if $P \wedge Q$ holds, so does $Q \wedge P$.

$$\dfrac{\dfrac{\dfrac{}{P \wedge Q \vdash P \wedge Q}}{P \wedge Q \vdash Q}\ \wedge\text{-E}_2 \qquad \dfrac{\dfrac{}{P \wedge Q \vdash P \wedge Q}}{P \wedge Q \vdash P}\ \wedge\text{-E}_1}{\dfrac{P \wedge Q \vdash Q \wedge P}{\vdash P \wedge Q \Rightarrow Q \wedge P}\ \text{I}\Rightarrow}\ \wedge\text{-I}$$

Example: $(P \wedge P) \Rightarrow P$

Question

Give a closed natural deduction proof of $(P \wedge P) \Rightarrow P$.

Example: $(P \wedge P) \Rightarrow P$

Question

Give a closed natural deduction proof of $(P \wedge P) \Rightarrow P$.

$$\dfrac{\dfrac{\overline{P \wedge P \vdash P \wedge P}}{P \wedge P \vdash P} \wedge E_1}{\vdash (P \wedge P) \Rightarrow P} \Rightarrow -I$$

Is this the only such proof?

Wrong proofs

The statement $(P \Rightarrow P) \Rightarrow P$ is not true in general.

We previously saw how we 'abused' proof strategies to come up with an incorrect proof.

What kind of mistakes can we make when we writing a proof using natural deduction?

Wrong proofs

The statement $(P \Rightarrow P) \Rightarrow P$ is not true in general.

We previously saw how we 'abused' proof strategies to come up with an incorrect proof.

What kind of mistakes can we make when we writing a proof using natural deduction?

$$\cfrac{\overline{P \Rightarrow P \vdash P}}{\vdash (P \Rightarrow P) \Rightarrow P} \Rightarrow\!-I$$

Here we can make the previous mistake more explicit: we are using the assumption P, whereas we can only use the assumption $P \Rightarrow P$.

Implication elimination

Proof of $P \Rightarrow Q$.

Proof of P.

Therefore, we can conclude Q $\square$.

Question

What is the rule for implication elimination?

Implication elimination

Proof of $P \Rightarrow Q$.

Proof of P.

Therefore, we can conclude Q $\square$.

Question

What is the rule for implication elimination?

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash P \Rightarrow Q}{\Gamma \vdash Q} \Rightarrow E$$

Natural deduction

We'll go through the rules for natural deduction for propositional logic.

Many of these rules closely mirror the proof strategies that we have seen previously – which is no coincidence of course.

They should be fairly familiar.

Once we've seen the rules for natural deduction proofs – we can try to relate them to the *truth table semantics* from our first lecture.

Truth and falsity

Most logic textbooks use $\top$ for **T** (truth) and $\bot$ for **F** (falsity).

The introduction rule for truth is trivial:

$$\frac{}{\Gamma \vdash \top}\ \top\text{-I}$$

There is no introduction rule for falsity.

Falsity elimination

Proof of a contradiction

Therefore we conclude P.

Or written as an inference rule:

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash P}\ \bot\text{-E}$$

Negation rules

Recall that $\neg P$ behaves just like $P \Rightarrow \bot$.

$$\frac{\Gamma \vdash \neg P \qquad \Gamma \vdash P}{\Gamma \vdash \bot} \neg\text{-E}$$

$$\frac{\Gamma, P \vdash \bot}{\Gamma \vdash \neg P} \neg\text{-I}$$

These two rules are really just instances of the rules for $P \Rightarrow Q$, where Q is taken to be $\bot$.

Equivalence rules

Similarly, $P \Leftrightarrow Q$ behaves the same as $P \Rightarrow Q \wedge Q \Rightarrow P$.

$$\frac{\Gamma, P \vdash Q \qquad \Gamma, Q \vdash P}{\Gamma \vdash P \Leftrightarrow Q} \Leftrightarrow\text{-I}$$

$$\frac{\Gamma \vdash P \Leftrightarrow Q}{\Gamma \vdash P \Rightarrow Q} \Leftrightarrow\text{-}E_1$$

$$\frac{\Gamma \vdash P \Leftrightarrow Q}{\Gamma \vdash Q \Rightarrow P} \Leftrightarrow\text{-}E_2$$

Exercise

Question

Prove that $\vdash P \Rightarrow (Q \Rightarrow (Q \wedge P))$

Exercise

Question

Prove that $\vdash P \Rightarrow (Q \Rightarrow (Q \land P))$

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{P, Q \vdash Q} \qquad \dfrac{}{P, Q \vdash P}}{P, Q \vdash Q \land P} \land\text{-I}}{P \vdash Q \Rightarrow (Q \land P)} \Rightarrow\text{-I}}{\vdash P \Rightarrow (Q \Rightarrow (Q \land P))} \Rightarrow\text{-I}
$$

Exercise

Question

Prove that $\vdash P \wedge \top \Leftrightarrow P$.

Exercise

Question

Prove that $\vdash P \wedge \top \Leftrightarrow P$.

$$\dfrac{\dfrac{\dfrac{}{P \vdash P} \qquad \dfrac{}{P \vdash \top}\ \top\text{-I}}{P \vdash P \wedge \top}\ \wedge\text{-I} \qquad \dfrac{\dfrac{}{P \wedge \top \vdash P \wedge \top}}{P \wedge \top \vdash P}\ \wedge\text{-E}_1}{\vdash P \wedge \top \Leftrightarrow P}\ \Leftrightarrow\text{-I}$$

What's missing?

The only thing remaining are the rules for disjunction.

The *introduction* rules are easy:

What's missing?

The only thing remaining are the rules for disjunction.

The *introduction* rules are easy:

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \vee\text{-}I_1$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \vee\text{-}I_2$$

Disjuction elimination: proof strategy

Proof of $P \vee Q$

Assume that P is true.

Proof of R

Next, assume Q is true.

Proof of R

Therefore, R is true, regardless of which of P or Q is true.

Disjuction elimination

$$\frac{\Gamma \vdash P \vee Q \qquad \Gamma, P \vdash R \qquad \Gamma, Q \vdash R}{\Gamma \vdash R} \vee\text{-E}$$

If we know $P \vee Q$ holds...

... and we know that R holds whenever P does;

... and we know that R holds whenever Q does;

... we can conclude that R must always hold.

Exercise

Question

Give a proof that $\vdash (P \vee \perp) \Rightarrow P$.

Exercise

Question

Give a proof that $\vdash (P \vee \bot) \Rightarrow P$.

$$\dfrac{\dfrac{\dfrac{}{P \vee \bot \vdash P \vee \bot} \qquad \dfrac{}{P \vee \bot, P \vdash P} \qquad \dfrac{\dfrac{}{P \vee \bot, \bot \vdash \bot}}{P \vee \bot, \bot \vdash P}}{P \vee \bot \vdash P}\ \vee\text{-E}}{\vdash P \vee \bot \Rightarrow P}\ \Rightarrow\text{-I}$$

Final rules

We need one final rule:

$$\frac{\Gamma, \neg P \vdash \bot}{\Gamma \vdash P} \text{ RAA}$$

This rule, sometimes called *reductio ad absurdum*, states that if $\neg P$ leads to a contradiction, P must hold.

'If it is impossible for ¬P to hold, P must hold.'

(Notice how it is the only rule that is not an introduction-elimination rule for a logical operator?)

Natural deduction for propositional logic

This completes the proof rules for natural deduction proofs in propositional logic.

For each logical operator, we have an *introduction* rule (stating how to prove a statement of a certain form) and an *elimination* rule (stating how to use a statement of a certain form).

There are many exercises to practice with in the lecture notes – **there will be a similar question on the exam**.

Natural deduction for propositional logic

This completes the proof rules for natural deduction proofs in propositional logic.

For each logical operator, we have an *introduction* rule (stating how to prove a statement of a certain form) and an *elimination* rule (stating how to use a statement of a certain form).

There are many exercises to practice with in the lecture notes – **there will be a similar question on the exam**.

There is also an online tutorial here:

`https://ics-websites.science.uu.nl/docs/vakken/b1li/holbert/`

Natural deduction for propositional logic

Are these natural deduction proofs ‘the same’ as the truth table semantics we saw previously?

Natural deduction for propositional logic

Are these natural deduction proofs ‘the same’ as the truth table semantics we saw previously?

To answer that question, we need to make our truth table semantics a bit more precise.

Semantics of propositional logic

When we fill out a truth table for some propositional formula *p*, we show how each choice of atomic propositional variables of *p* results in a true/false value.

$\mathtt{p}$	$\mathtt{q}$	$\neg$	$(\mathtt{p}$	$\vee$	$\mathtt{q})$	$\Rightarrow$	$(\neg\mathtt{p}$	$\wedge$	$\neg\mathtt{q})$
F	F	T	F	F	F	**T**	T	T	T
F	T	F	F	T	T	**T**	T	F	F
T	F	F	T	T	F	**T**	F	F	T
T	T	F	T	T	T	**T**	F	F	F

For each value of $\mathtt{p}$ and $\mathtt{q}$, we can check the corresponding row to see the value of the entire proposotional formula.

Can we make this more precise?

Semantics of propositional logic

Let's try to define a function by induction over propositional logic formulas, mapping each formula to a boolean value.

Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid \neg p \mid P \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

Semantics of propositional logic

Let's try to define a function by induction over propositional logic formulas, mapping each formula to a boolean value.

Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid \neg p \mid P \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

- if p is true, we return **T**;

Semantics of propositional logic

Let's try to define a function by induction over propositional logic formulas, mapping each formula to a boolean value.

Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid \neg p \mid P \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

- if p is true, we return **T**;
- if p is false, we return **F**;

Semantics of propositional logic

Let's try to define a function by induction over propositional logic formulas, mapping each formula to a boolean value.

Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid \neg p \mid P \mid p \land q \mid p \lor q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

- if p is true, we return **T**;
- if p is false, we return **F**;
- if p is of the form $\neg q$, we can compute the value associated with q. If this is **T**, we return **F**; if it is **F**, we return **T**.

Semantics of propositional logic

Let's try to define a function by induction over propositional logic formulas, mapping each formula to a boolean value.

Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid \neg p \mid P \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

- if p is true, we return **T**;
- if p is false, we return **F**;
- if p is of the form $\neg q$, we can compute the value associated with q. If this is **T**, we return **F**; if it is **F**, we return **T**.
- But what do we do if our input formula p is an atomic propositional variable P? We don't 'know' whether P is **T** or **F**?

Semantics of propositional logic

We call a function $v : \mathbf{P} \longrightarrow \mathbf{Bool}$ a *truth assignment*.

Such a function tells us the values of associated with each atomic propositional variables.

Claim Given any truth assignment v and propositional logic formula p, we **can** calculate the truth value of a p.

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \longrightarrow \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

We can do this by induction on p. Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \longrightarrow \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

We can do this by induction on p. Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

- if p is true, we return **T**;
- if p is false, we return **F**;
- if p is of the form $\neg q$, we can compute the value associated with q. If this is **T**, we return **F**; if it is **F**, we return **T**.

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \rightarrow \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

We can do this by induction on p. Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \rightarrow \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

We can do this by induction on p. Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

- if p is an atomic propositional variable $P \in \mathbf{P}$, we return $v(P)$.

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \rightarrow \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

We can do this by induction on p. Recall that the propositional logic formulas are given by the following BNF:

$$p, q \quad ::= \quad \text{true} \mid \text{false} \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

- if p is an atomic propositional variable $P \in \mathbf{P}$, we return $v(P)$.
- if p is of the form $q_1 \wedge q_2$, we can compute the value associated with q_1 and q_2. If this both are **T**, we return **T**; otherwise we return **F**.
- if p is of the form $q_1 \vee q_2$, we can compute the value associated with q_1 and q_2. If this both are **F**, we return **F**; otherwise we return **T**.
- similar cases exist for implication and logical equivalence.

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \longrightarrow \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

Our truth assignment tells us exactly how to treat atomic propositions.

For each of the logical operators, such as conjunction, we can define their behaviour by taking the corresponding operation on booleans, such as the boolean and.

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \to \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

This defines the semantics of all propositional logic formulas, usually written $[\![p]\!]$.

$$[\![p]\!] : (\mathbf{P} \to \mathbf{Bool}) \to \mathbf{Bool}$$

That is, we have defined a function that maps each propositional logic formula p into a function that, given a truth assignment for all atomic propositional variables, computes the truth value of the entire propositional logic formula p.

Semantics of propositional logic

Claim Given any truth assignment $v : \mathbf{P} \rightarrow \mathbf{Bool}$ and propositional logic formula p, we can calculate the truth value of a p.

This defines the semantics of all propositional logic formulas, usually written $[\![p]\!]$.

$$[\![p]\!] : (\mathbf{P} \rightarrow \mathbf{Bool}) \rightarrow \mathbf{Bool}$$

That is, we have defined a function that maps each propositional logic formula p into a function that, given a truth assignment for all atomic propositional variables, computes the truth value of the entire propositional logic formula p.

But what does this have to do with truth tables?

Finite functions

If you think back to the lectures on functions and induction, we saw how to *define* a function on a *finite* domain by listing all it output value for every possible input value.

Suppose I'm teaching a class with 5 students

S = $\{$Alice, Bob, Carroll, David, Eve $\}$.

I can define a functions `marks` mapping $S \rightarrow \{1..10\}$ by giving each student their mark:

marks(Alice) = 8

marks(Bob) = 6

marks(Carroll) = 7

...

Finite functions and truth tables

When filling out a truth table for some propositional logic formula p, you are essentially computing the truth value of *p for all possible choice of value for the atomic variables in p*.

For any formula p, there are $2^{|fv(p)|}$ possible truth assignments for the free variables in p.

Hence, you can give the semantics for p, that is the function:

$$[\![p]\!] : (\mathbf{P} \to \mathbf{Bool}) \to \mathbf{Bool}$$

as a truth table with $2^{|fv(p)|}$ rows.

Truth tables are simply the tabulation of this semantics.

Natural deduction vs semantics

Given any propositional logic formula p, we can assign it semantics:

$$[\![p]\!] : (\mathbf{P} \to \mathbf{Bool}) \to \mathbf{Bool}$$

But how is this semantics related to our natural deduction rules?

Our inference rules for natural deduction all seem perfectly ‘logical’.

But can we be sure that any propositional formula proven using this inference rules always holds?

And can we be sure that we haven’t left out any inference rules?

Notation

Given a set of propositional logic formulas, Γ, we will write $\Gamma \vdash p$ whenever we can find a natural deduction proof of the formula p using the assumptions from Γ.

When we do not need any assumptions to show p, we write $\vdash p$.

Notation

Given a set of propositional logic formulas, Γ, we will write $\Gamma \vdash p$ whenever we can find a natural deduction proof of the formula p using the assumptions from Γ.

When we do not need any assumptions to show p, we write $\vdash p$.

Given an truth assignment v we write $v \models p$ if $[\![p]\!](v) = \mathbf{T}$.

If for all truth assignments v, we have $v \models p$ we say that $\models p$ (and p is a tautotology).

Soundness and completeness

It turns out that natural deduction inference rules above satisfy two important properties:

Soundness If $\vdash p$ then $\models p$. In other words, if we can find a proof of p using the inference rules of natural deduction, then the truth table of p consists of only **T**.

Completeness If $\models p$ then $\vdash p$. In other words, if the truth table of p consists of only **T**, there is *some* derivation of p using the inference rules of natural deduction.

Proofs?

The proofs of soundness and completeness are a subject of a more advanced course on formal logic. . .

. . . but in principle you have the reasoning techniques to understand them.

I will sketch both proofs briefly.

Soundness

Soundness is relatively easy to show: given a derivation of some formula *p*, we can do induction on this derivation. If we can show each of our inference rules is safe to use, we can trust each proof built using them.

For example, compare the natural deduction rule:

$$\frac{P \qquad Q}{P \wedge Q} \wedge\text{-I}$$

And the following line from the truth table for conjunction:

p	q	(p	∧	q)
T	T	T	T	T

Soundness

We should check each of the rules of natural deduction to establish that:

in each row of the truth table where premises are true, the conclusion is also true

Soundness

We should check each of the rules of natural deduction to establish that:

in each row of the truth table where premises are true, the conclusion is also true

In short, there is work to be done – but it is a simple check for each inference rule of natural deduction.

Completeness

Completeness is harder: we don't have a derivation – we cannot just inspect the rules of natural deduction.

Instead we need to create a derivation for some arbitrary formula p...

The only thing we know about p is that it is a tautology in the truth table semantics.

Completeness

Completeness is harder: we don't have a derivation – we cannot just inspect the rules of natural deduction.

Instead we need to create a derivation for some arbitrary formula p...

The only thing we know about p is that it is a tautology in the truth table semantics.

How can we find a derivation?

Completeness

The idea is to perform induction on the number of atomic propositional variables in our formula p.

If p has no variables, it must be (equivalent to) **T** – since it is true according to our truth table semantics.

If p has at least one variable P, we can construct a derivation as follows:

- we can prove $P \vee \neg P$ holds (exercise);
- we can show that $P \vdash p$ (induction hypothesis on the top half of the truth table);
- we can show that $\neg P \vdash p$ (induction hypothesis on the bottom half of the truth table);

Using the disjunction elimination rule we can prove that p must hold.

Soundness and completeness

These results show just how clean and simple propositional logic is...

But they break down as soon as you study richer predicate logics...

Kurt Gödel

Kurt Gödel

Gödel's incompleteness theorems

Kurt Gödel famously showed an important **incompleteness** result.

Gödel's incompleteness theorems

Kurt Gödel famously showed an important **incompleteness** result.

Any consistent formal system F within which a certain amount of elementary arithmetic can be carried out is incomplete; i.e., there are statements of the language of F which can neither be proved nor disproved in F.

In other words, there is **no** complete set of inference rules for more interesting logics that support elementary arithmetic.

How on earth do you prove this?

Gödel’s incompleteness theorems

Kurt Gödel famously showed an important **incompleteness** result.

Any consistent formal system F within which a certain amount of elementary arithmetic can be carried out is incomplete; i.e., there are statements of the language of F which can neither be proved nor disproved in F.

In other words, there is **no** complete set of inference rules for more interesting logics that support elementary arithmetic.

How on earth do you prove this?

Gödel managed to show how to write “This statement has no derivation”.

- If this statement does have a derivation, our logic is unsound (and we can prove falsity).
- But if our logic is sound and the statement holds, no derivation can exist…

Material

Chapter 1-2 of the lecture notes.