Logic for Computer Science

10 – Proofs by induction

Paige Randall North
(based on Wouter Swierstra's slides)

University of Utrecht

Last time

Inductive definitions

This lecture

Proofs by induction

Organization

- 'Fun' lecture on Thursday on the Lean proof assistant – and using it to verify the kinds of proofs we have already seen.
- Use this week to *catch up* – don't just focus on recovering from the mid-term! Make sure that you are ready for the first quiz after the break.
- The lectures on induction are *very important* – not just for this course, but this material will show up again and again throughout the rest of your degree.

Previously: definitions using induction

In the last lecture, we studied how to give an *inductive* definition of a set, function or relation.

For example, we can define the set of natural numbers $\mathbb{N}$ as follows:

- $0 \in \mathbb{N}$
- for any $n \in \mathbb{N}$, the number $n + 1 \in \mathbb{N}$.
- there are no other elements of $\mathbb{N}$.

Example: sum

We can then define a function over $\mathbb{N}$ by induction.

For example, we may want to compute the sum of the first n numbers:

$$1 + 2 + 3 + \ldots + n$$

We can do so using an inductive definition:

$$\begin{aligned} \mathtt{sum}(0) &= 0 \\ \mathtt{sum}(k+1) &= (k+1) + \mathtt{sum}(k) \end{aligned}$$

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

How to prove this? Let's check that the equality holds for the first few numbers:

- if $n = 0$, we have that $\texttt{sum}(0) = 0 = \frac{0(0+1)}{2}$.

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

How to prove this? Let's check that the equality holds for the first few numbers:

- if $n = 0$, we have that $\texttt{sum}(0) = 0 = \frac{0(0+1)}{2}$.
- if $n = 1$, we have that $\texttt{sum}(1) = 0 + 1 = 1 = \frac{1(1+1)}{2}$.

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

How to prove this? Let's check that the equality holds for the first few numbers:

- if $n = 0$, we have that $\texttt{sum}(0) = 0 = \frac{0(0+1)}{2}$.
- if $n = 1$, we have that $\texttt{sum}(1) = 0 + 1 = 1 = \frac{1(1+1)}{2}$.
- if $n = 2$, we have that $\texttt{sum}(2) = 0 + 1 + 2 = 3 = \frac{2(2+1)}{2}$.

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

How to prove this? Let's check that the equality holds for the first few numbers:

- if $n = 0$, we have that $\texttt{sum}(0) = 0 = \frac{0(0+1)}{2}$.
- if $n = 1$, we have that $\texttt{sum}(1) = 0 + 1 = 1 = \frac{1(1+1)}{2}$.
- if $n = 2$, we have that $\texttt{sum}(2) = 0 + 1 + 2 = 3 = \frac{2(2+1)}{2}$.
- if $n = 3$, we have that $\texttt{sum}(3) = 0 + 1 + 2 + 3 = 6 = \frac{3(3+1)}{2}$.

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

How to prove this? Let's check that the equality holds for the first few numbers:

- if $n = 0$, we have that $\texttt{sum}(0) = 0 = \frac{0(0+1)}{2}$.
- if $n = 1$, we have that $\texttt{sum}(1) = 0 + 1 = 1 = \frac{1(1+1)}{2}$.
- if $n = 2$, we have that $\texttt{sum}(2) = 0 + 1 + 2 = 3 = \frac{2(2+1)}{2}$.
- if $n = 3$, we have that $\texttt{sum}(3) = 0 + 1 + 2 + 3 = 6 = \frac{3(3+1)}{2}$.

But when are we done? We need a proof!

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

Example: sums

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

Proof

According to the proof strategy for universal quantification we saw previously, we assume k is a number.

Now we need to show that $\texttt{sum}(k) = \frac{k \times (k+1)}{2} \ldots$

But now we're stuck – we don't know anything about $\texttt{sum}(k)$ for some arbitrary k.

We need more advanced proof techniques than the ones we have seen so far.

Proofs by induction - a new strategy

We defined the set of natural numbers using the following two clauses:

- $0 \in \mathbb{N}$
- for any $n \in \mathbb{N}$, the number $(n + 1) \in \mathbb{N}$.

Proofs by induction - a new strategy

We defined the set of natural numbers using the following two clauses:

- $0 \in \mathbb{N}$
- for any $n \in \mathbb{N}$, the number $(n + 1) \in \mathbb{N}$.

To show that some property P holds for all natural numbers, it suffices to show:

- $P(0)$
- for all k, if we assume that $P(k)$ we need to show that $P(k + 1)$

Example proof by induction

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

Example proof by induction

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

Proof:

We prove this statement by induction on n.

Example proof by induction

Claim: For all n, we can show that

$$\texttt{sum}(n) = \frac{n(n+1)}{2}.$$

Proof:

We prove this statement by induction on n.

- If $n = 0$, we need to show that $\texttt{sum}(0) = \frac{0(0+1)}{2}$.
- Suppose that $n = k + 1$ and that

  $$\texttt{sum}(k) = \frac{k(k+1)}{2}.$$

  We need to show that

  $$\texttt{sum}(k+1) = \frac{(k+1)(k+2)}{2}.$$

We'll do both steps separately.

Example: base case

If $n = 0$, we need to show that $\texttt{sum}(0) = \frac{0(0+1)}{2}$.

Using the definition of $\texttt{sum}$, we know that $\texttt{sum}(0) = 0 = \frac{0(0+1)}{2}$ as required.

This completes the base case.

Example: inductive case

Suppose that that $\texttt{sum}(k) = \frac{k \times (k+1)}{2}$.

We need to show $\texttt{sum}(k+1) = \frac{(k+1)(k+2)}{2}$:

$$
\begin{aligned}
\texttt{sum}(k+1) &= (k+1) + \texttt{sum}(k) && \text{by definition of } \texttt{sum} \\
&= (k+1) + \frac{k(k+1)}{2} && \text{by our } \textit{induction hypothesis} \\
&= (k+1)\left(1 + \frac{k}{2}\right) && \text{distributivity} \\
&= (k+1)\left(\frac{k+2}{2}\right) && \text{arithmetic} \\
&= \frac{(k+1)(k+2)}{2} && \text{arithmetic}
\end{aligned}
$$

Hence we have established the inductive step as required.

Proofs by induction

Why does this work?

- The base case tells us that $P(0)$
- The inductive step shows that if $P(n)$ then we can conclude $P(n+1)$.

So:

- The base case states that our theorem is true for 0, that is $P(0)$ holds.
- Using our induction hypothesis, we can prove the that the theorem holds for 1, that is $P(1)$ holds.
- Using our induction hypothesis again, we can prove the that the theorem holds for 2, that is $P(2)$ holds.
- Using our induction hypothesis again, we can prove the that the theorem holds for 3, that is $P(3)$ holds.
- ...

Proofs by induction

In this fashion we can construct a proof that our theorem holds for *any* number.

People sometimes make an analogy with dominos:

- The base case ensures that the first stone will fall;
- The step case ensures that each domino will knock down the next.

By repeatedly applying the inductive step to the base case, we can construct the desired proof for any number.

Hence, we can conclude that for all n, our property $P(n)$ holds.

Finite induction

Checking a statement about infinitely many numbers in a finite number of cases is **not** a proof.

The book explains several famous counter-examples, such as the *Fermat numbers*:

$$F_n := 2^{2^n} + 1$$
$$F_0 = 3$$
$$F_1 = 5$$
$$F_2 = 17$$
$$F_3 = 257$$
$$F_4 = 65,537\ldots$$

It seems like this is a reliable way to generate increasingly large prime numbers!

Finite induction

But it turns out that $F_5 = 4,294,967,297 = 641 \cdot 6,700,417$.

This example demonstrates that we cannot draw conclusions about *all* numbers based on a *finite* collection of tests.

This is precisely what proofs by induction do guarantee!

A proof by induction gives you a reliable (finite) 'recipe' for constructing proofs for any number.

Another example

Question

For any natural number $n \geq 8$, prove that we can write n as $3x + 5y$, for some natural numbers x and y.

Prove this by induction on n.

Another example

Question

For any natural number $n \geq 8$, prove that we can write n as $3x + 5y$, for some natural numbers x and y.

Prove this by induction on n.

Proof: We proceed by induction on n:

- if $n = 8$, we can write it as $3 \times 1 + 5 \times 1$.
- if n is larger then 8, assume we can write it as $3x + 5y$ for some x and y.

Another example

Question

For any natural number $n \geq 8$, prove that we can write n as $3x + 5y$, for some natural numbers x and y.

Prove this by induction on n.

Proof: We proceed by induction on n:

- if $n = 8$, we can write it as $3 \times 1 + 5 \times 1$.
- if n is larger then 8, assume we can write it as $3x + 5y$ for some x and y.

We now need to show how to write $n + 1$ in this form. We distinguish two cases:

- if $y = 0$, then we can write $n + 1 = 3x + 1 = 3(x - 3) + 5 \cdot 2$.*
- if $y > 0$, then we can write $n + 1 = 3x + 5y + 1 = 3(x + 2) + 5(y - 1)$.

About this proof

Note that this proof doesn't follow the exact same recipe.

Instead we show:

- $P(8)$
- if $P(k)$ then $P(k+1)$

This doesn't show that $P(n)$ holds for all n, but does show that it holds for all $n \geq 8$.

Example: fibonacci numbers

Prove the following theorem by induction on n.

For any number n, the following equality holds on Fibonacci numbers:

$$f_0 + f_1 + f_2 + \ldots + f_n = f_{n+2} - 1.$$

Proof:

- If $n = 0$, then $f_0 + f_1 + \ldots + f_0 = f_0 = 0$. While $f_{0+2} - 1 = f_2 - 1 = 1 - 1 = 0$. Hence the left and right hand sides are equal.
- Suppose the equation holds for k. We have:

$$\begin{aligned} f_0 + f_1 + f_2 + \ldots + f_k + f_{k+1} &= (f_{k+2} - 1) + f_{k+1} && \text{(induction hypothesis)} \\ &= (f_{k+1} + f_{k+2}) - 1 && \text{(arithmetic)} \\ &= f_{k+3} - 1 && \text{(definition of Fibonacci)} \end{aligned}$$

Strong induction

The induction principle on natural numbers states that to show some property P holds for all natural numbers, it suffices to show:

- $P(0)$
- $\forall k,\ P(k) \Rightarrow P(k+1)$.

In particular, in the inductive case, the induction hypothesis *only* holds for $P(k)$.

This is fine for many examples, but what if we need to assume that P holds not just for the *previous* number, but rather for *all* previous numbers?

Example: strong induction

Consider the following function $f : \mathbb{N} \to \mathbb{N}$:

$$f(n) = \begin{cases} 0 & \text{when } n = 0 \\ 2 \cdot f(n/2) & \text{when } n \text{ is even} \\ f(n-1) + 1 & \text{when } n \text{ is odd} \end{cases}$$

Lemma For all n, $f(n) = n$.

Example: strong induction

Consider the following function $f : \mathbb{N} \to \mathbb{N}$:

$$f(n) = \begin{cases} 0 & \text{when } n = 0 \\ 2 \cdot f(n/2) & \text{when } n \text{ is even} \\ f(n-1) + 1 & \text{when } n \text{ is odd} \end{cases}$$

Lemma For all n, $f(n) = n$.

We use *strong induction* on n and distinguish three cases:

- if $n = 0$, then by definition $f(0) = 0$
- if n is even, then $f(n) = 2 \cdot f(n/2) = 2 \cdot (n/2) = n$
- if n is odd, $f(n) = f(n-1) + 1 = n - 1 + 1 = n$

Strong induction

Question

Which of the following cases is not provable using regular induction?

Example: strong induction

We use strong induction on n and distinguish three cases:

- if $n = 0$, then by definition $f(0) = 0$
- if n is even, then $f(n) = 2 \times f(n/2) = 2 \times (n/2) = n$
- if n is odd, $f(n) = f(n-1) + 1 = n - 1 + 1 = n$

Beyond natural numbers

We can perform induction on other inductively defined sets, such as binary trees and lists.

Recall that binary trees are defined as follows:

$$t ::= \star \mid N(t_1, t_2)$$

What induction principle would we expect?

Induction on trees

- If $P(\star)$ holds – that is P holds for each leaf $\star$;
- And if we can conclude that $P(N(t_1, t_2))$ holds, provided $P(t_1)$ and $P(t_2)$.

Then we can conclude that $P(t)$ holds for every binary tree t.

Once again, the *inductive* structure of the set we have defined determines the *induction principle* that we can use to reason about trees.

Example: induction on trees

Theorem: For every binary tree t, there is exactly one more leaf in t than there are internal nodes.

Example: induction on trees

Theorem: For every binary tree t, there is exactly one more leaf in t than there are internal nodes.

Proof Proceed by induction on the tree t.

- If $t = \star$, then there is 1 leaf and 0 nodes – hence our property holds;
- If $t = N(t_1, t_2)$ then by induction we know that for both t_1 and t_2 there is one more leaf than there is node. Hence by constructing the composite tree $N(t_1, t_2)$ we have added one internal node, but the total number of leaves is still one greater than the number of nodes.

Lists

We defined lists of numbers as:

$$L ::= [\,] \mid n : L$$

Each list is either:

- equal to the empty list [ ] that has no elements in it;
- or consists of two parts:
  - a first element n stored at the *head* of the list;
  - the remainder (or *tail*) of the list.

What induction principle would you expect on lists?

Induction lists

To show that some property P holds for every list L, it suffices to show that:

- $P([\,])$ – that is P holds for the empty list;
- assuming that $P(L)$, we need to show that for all n, that $P(n : L)$ also holds.

Example: induction on lists

Theorem: The append function is associative. That is, for all lists xs, ys and zs we have that

$$\mathtt{append}(xs, \mathtt{append}(ys, zs)) = \mathtt{append}(\mathtt{append}(xs, ys), zs)$$

where append is defined by

$$\mathtt{append}([\,], ys) = ys$$
$$\mathtt{append}(n : xs, ys) = n : \mathtt{append}(xs, ys)$$

Exercise

Prove this by induction.

Propositional formulas

We can describe the structure of propositional logic formulas as follows:

$$p, q ::= \top \mid \bot \mid P \mid \neg p \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid p \Leftrightarrow q$$

This definition gives rise to an induction principle. To prove some property R holds for all propositional formulas, we need to check:

- $R(\top)$ and $R(\bot)$;
- $R(P)$ for all P;
- and R is preserved by all the logical operators in the usual fashion.

Example: induction on propositional formulas

We can define the NAND operation as follows:

p	q	$p \mid q$
F	**F**	**T**
F	**T**	**T**
T	**F**	**T**
T	**T**	**F**

Claim: Every propositional logical formula is equivalent to a formula that exclusively uses the NAND operator, the constants $\top$ and $\bot$, and atomic formulas P.

Proof? By induction on the formula.

Example: induction on propositional formulas

p	q	$p \mid q$
F	**F**	**T**
F	**T**	**T**
T	**F**	**T**
T	**T**	**F**

For example, in the case for $\neg p$, we need to show how to implement an equivalent formula using the NAND operator.

In this case, we can implement $\neg p$ as $p \mid p$.

Example: induction on propositional formulas

p	q	$p \mid q$
F	**F**	**T**
F	**T**	**T**
T	**F**	**T**
T	**T**	**F**

For example, in the case for $p \wedge q$, we need to show how to implement an equivalent formula using the NAND operator.

In this case, we can implement $p \wedge q$ as $\neg(p \mid q)$ or rather, $(p \mid q) \mid (p \mid q)$.

To complete the proof, we need to find similar formulations for disjunction, implication, and equivalence.

Recall that we defined addition as:

$$\texttt{add}(0, n) = n$$
$$\texttt{add}(k + 1, n) = \texttt{add}(k, n) + 1$$

We can now prove properties of addition by induction!

Lemma For all n, $\texttt{add}(n, 0) = n$.

Recall that we defined addition as:

$$\texttt{add}(0, n) = n$$
$$\texttt{add}(k + 1, n) = \texttt{add}(k, n) + 1$$

We can now prove properties of addition by induction!

Lemma For all n, $\texttt{add}(n, 0) = n$.

Proof Induction on n...

Inductions gone wrong

Although this seems obvious enough, there are plenty of examples of where induction cannot be used or is used incorrectly.

Let's take a closer look at a few examples...

Grains of sand

'Theorem' For all $n \geq 0$, n grains of sand do not make up a sandpile.

'Proof'

Clearly when $n = 0$, we do not have a sandpile.

Suppose we have $n + 1$ grains of sand. Assume that these *do* make up a sandpile. Removing one grain of sand from a sandpile will still leave us with a sandpile. But by our induction hypothesis, we cannot form a sandpile with n grains of sand. Therefore we cannot construct a sandpile with $n + 1$ grains of sand.

Question

What is wrong with the above argument?

Grains of sand

'Theorem' For all $n \geq 0$, n grains of sand do not make up a sandpile.

'Proof'

Clearly when $n = 0$, we do not have a sandpile.

Suppose we have $n + 1$ grains of sand. Assume that these *do* make up a sandpile. Removing one grain of sand from a sandpile will still leave us with a sandpile. But by our induction hypothesis, we cannot form a sandpile with n grains of sand. Therefore we cannot construct a sandpile with $n + 1$ grains of sand.

Question

What is wrong with the above argument?

It relies heavily on a poorly defined notion of 'sandpile' – without a precise definition, we cannot reason in this style.

Another example

Consider the following function $T : \mathbb{Z} \to \mathbb{Z}$:

$$T(n) = \begin{cases} n + 6 & \text{when } n \leq 0 \\ T(T(n-7)) & \text{otherwise} \end{cases}$$

Lemma $T(n) = 6$ for all $n \geq 0$.

Proof Induction on n:

- If $n = 0$, then $T(0) = 6$ as expected;
- If $n > 0$, then $T(n) = T(T(n-7)) = T(6) = 6$ (by applying our induction hypothesis twice).

Another example

Consider the following function $T : \mathbb{Z} \to \mathbb{Z}$:

$$T(n) = \begin{cases} n + 6 & \text{when } n \leq 0 \\ T(T(n-7)) & \text{otherwise} \end{cases}$$

Lemma $T(n) = 6$ for all $n \geq 0$.

Proof Induction on n:

- If $n = 0$, then $T(0) = 6$ as expected;
- If $n > 0$, then $T(n) = T(T(n-7)) = T(6) = 6$ (by applying our induction hypothesis twice).

But we don't know that $T(n-7) = 6$ or that $T(6) = 6$! (Why?)

Induction on relations

Besides inductively defined sets and functions, we also mentioned inductively defined *relations* in the previous lecture – such as less-than on natural numbers, or sortedness of lists.

What induction principle is associated with such inductively defined relations?

Rule induction

Given a proof that $x \leq y$, we can perform induction on three things:

- the number x;
- the number y;
- the *proof* that $x \leq y$

We know how to perform induction on numbers - but how can we perform induction on proofs?

Rule induction

Given a proof that $x \leq y$, we can perform induction on three things:

- the number x;
- the number y;
- the *proof* that $x \leq y$

We know how to perform induction on numbers - but how can we perform induction on proofs?

This requires a more advanced technique, sometimes referred to as *rule induction*.

Rule induction

We defined the $\leq$ *relation* between natural numbers using the following rules:

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

Any proof that $x \leq y$ is constructed using these rules.

Hence if we have a proof $x \leq y$, we can distinguish two cases:

- either the proof was built using the zero-rule, in which case $x = 0$ and the proof we have states $0 \leq y$;
- or the proof was built using the succ-rule, in which case $x = s(n)$ and $y = s(m)$ - that is, x and y are non zero - and we have a 'smaller' proof that $n \leq m$.

Rule induction

Lemma If $x \leq y$ and $y \leq z$ then $x \leq z$.

Proof Use rule induction on our assumptions:

By rule induction on the first proof:

- if it is built using the zero-rule, then $x = 0$ and hence we can show $0 \leq z$ as required.
- if it is built using the succ-rule, then $y = s(y')$ and $x = s(x')$ and we have a 'smaller' proof that $x' \leq y'$. If we consider the possible proofs of $y \leq z$, this must also be constructed using the succ-rule as y is non-zero, so we know that $z = s(z')$ for some z'. Hence all of x, y and z are necessarily non-zero and we know that $y' \leq z'$. By induction on the proofs, we can conclude that $x' \leq z'$ and using the succ-rule we can construct the desired proof that $x \leq z$.

Rule induction

Such derivations using rule induction can become complex quickly!

Fortunately, proof assistants are *very good* at doing this kind of bookkeeping for us.

Many results about realistic programming languages and systems have been formally proven in such proof assistants – including a correctness proof of a realistic C compiler or important research results in mathematics.

Recap

In the last lecture, we studied how to define sets, functions, and relations using induction.

Recap

In the last lecture, we studied how to define sets, functions, and relations using induction.

In this lecture, we showed how to reason about these sets and functions using induction.

In each proof, we established that the base case was valid.

And showed that – assuming the property was true for smaller values – we could build a proof for some more complex structure.

Recap

The concept of induction is one of the cornerstones of computer science.

This allows you to define infinitely large sets with a finite description.

And to establish that a property of all inhabitants of an infinite set holds – in finite time!

It works for numbers, lists, trees, and even propositional logic itself!

Recap

The concept of induction is one of the cornerstones of computer science.

This allows you to define infinitely large sets with a finite description.

And to establish that a property of all inhabitants of an infinite set holds – in finite time!

It works for numbers, lists, trees, and even propositional logic itself!

This is mind-boggling!

Remaining weeks

In the remaining weeks, I want to wrap up the first part of the book: lecture on games after the Christmas break.

After that, we will cover some material that is not in the book – namely natural deduction, programming language semantics, and Hoare logic.

I have lecture notes on the website for these topics – if you want to have a look.

Material

- Modelling Computing Systems Chapter 9