Finite automata theory and formal languages (DIT321, TMV027)

Nils Anders Danielsson,
partly based on slides by Ana Bove

2019-01-24

Today

- Proofs.
- Induction for the natural numbers.
- Inductively defined sets.
- Recursive functions.

Some basic proof methods

Some basic proof methods

- To prove $p \Rightarrow q$, assume p and prove q.
- To prove $\forall x \in A.\ P(x)$, assume that we have an $x \in A$ and prove $P(x)$.
- To prove $p \Leftrightarrow q$, prove both $p \Rightarrow q$ and $q \Rightarrow p$.
- To prove $\neg p$, assume p and derive a contradiction.
- To prove p, prove $\neg\neg p$.
- To prove $p \Rightarrow q$, assume $\neg q$ and prove $\neg p$.

(There may be other ways to prove these things.)

Induction

Mathematical induction

For a natural number predicate P we can prove $\forall n \in \mathbb{N}.\ P(n)$ in the following way:

- Prove $P(0)$.
- For every $n \in \mathbb{N}$, prove that $P(n)$ implies $P(n+1)$.

With a formula:

$$P(0) \wedge (\forall n \in \mathbb{N}.\ P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \in \mathbb{N}.\ P(n)$$

Which of the following variants of induction are valid?

1. $P(0) \land (\forall n \in \mathbb{N}.\ n \geq 1 \land P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \in \mathbb{N}.\ n \geq 1 \Rightarrow P(n)$.
2. $P(1) \land (\forall n \in \mathbb{N}.\ n \geq 1 \land P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \in \mathbb{N}.\ n \geq 1 \Rightarrow P(n)$.
3. $P(1) \land P(2) \land (\forall n \in \mathbb{N}.\ n \geq 2 \land P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \in \mathbb{N}.\ n \geq 1 \Rightarrow P(n)$.

Counterexamples

- One can sometimes prove that a statement is invalid by using a counterexample.
- Example: The following statement does not hold for $P(n) := n \neq 1$ and $n = 1$:

  $$P(0) \land (\forall n \in \mathbb{N}.\ n \geq 1 \land P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \in \mathbb{N}.\ n \geq 1 \Rightarrow P(n)$$

  The hypotheses hold, but not the conclusion.

Counterexamples

More carefully:

- Let us prove

$$\neg(\forall \text{ natural number predicates } P.\ P(0) \wedge (\forall n \in \mathbb{N}.\ n \geq 1 \wedge P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \in \mathbb{N}.\ n \geq 1 \Rightarrow P(n)).$$

- We assume

$$\forall \text{ natural number predicates } P.\ P(0) \wedge (\forall n \in \mathbb{N}.\ n \geq 1 \wedge P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \in \mathbb{N}.\ n \geq 1 \Rightarrow P(n),$$

and derive a contradiction.

Counterexamples

- Let us use the predicate $P(n) := n \neq 1$.
- We have $P(0)$, i.e. $0 \neq 1$.
- We also have
  $\forall n \in \mathbb{N}.\ n \geq 1 \wedge P(n) \Rightarrow P(n+1)$, i.e.
  $\forall n \in \mathbb{N}.\ n \geq 1 \wedge n \neq 1 \Rightarrow n+1 \neq 1$.
- Thus we get $\forall n \in \mathbb{N}.\ n \geq 1 \Rightarrow P(n)$.
- Let us use $n = 1$.
- We have $1 \geq 1$.
- Thus we get $P(1)$, i.e. $1 \neq 1$.
- This is a contradiction, so we are done.

Complete induction

We can also prove $\forall n \in \mathbb{N}.\ P(n)$ in the following way:

- Prove $P(0)$.
- For every $n \in \mathbb{N}$, prove that if $P(i)$ holds for every natural number $i \leq n$, then $P(n+1)$ holds.

With a formula:

$$P(0) \wedge$$
$$(\forall n \in \mathbb{N}.\ (\forall i \in \mathbb{N}.\ i \leq n \Rightarrow P(i)) \Rightarrow P(n+1)) \Rightarrow$$
$$\forall n \in \mathbb{N}.\ P(n)$$

Which of the following variants of complete induction are valid?

1. $(\forall n \in \mathbb{N}.\ (\forall i \in \mathbb{N}.\ i < n \Rightarrow P(i)) \Rightarrow P(n)) \Rightarrow$
   $\forall n \in \mathbb{N}.\ P(n)$.

2. $P(1) \wedge$

   $$(\forall n \in \mathbb{N}.\ n \geq 1 \wedge (\forall i \in \mathbb{N}.\ i \leq n \Rightarrow P(i)) \Rightarrow P(n+1)) \Rightarrow$$

   $\forall n \in \mathbb{N}.\ P(n)$.

An example

Lemma

Every natural number $n \geq 8$ *can be written as a sum of multiples of* 3 *and* 5*.*

An example

Proof.

Let $P(n)$ be $n \geq 8 \Rightarrow \exists i, j \in \mathbb{N}.\ n = 3i + 5j$. We prove that $P(n)$ holds for all $n \in \mathbb{N}$ by complete induction on n:

- Base cases ($n = 0, ..., 7$): Trivial.
- Base cases ($n = 8$, $n = 9$, $n = 10$): Easy.
- Step case ($n \geq 10$, inductive hypothesis $\forall i \in \mathbb{N}. i \leq n \Rightarrow P(i)$, goal $P(n+1)$): Because $n - 2 \geq 8$ the inductive hypothesis for $n - 2$ implies that there are $i, j \in \mathbb{N}$ such that $n - 2 = 3i + 5j$. Thus we get $1 + n = 3 + (n - 2) = 3(i + 1) + 5j$. □

Proofs

How detailed should a proof be?

- Depends on the purpose of the proof.
- Who or what do you want to convince?
  - Yourself?
  - A fellow student?
  - An examiner?
  - An experienced researcher?
  - A computer program (a proof checker)?

Discuss the following proof of $\forall n \in \mathbb{N}.\ \sum_{i=0}^{n} i = n\frac{n+1}{2}$. Would you like to add/remove/change anything?

By induction on n:

- $n = 0$: $\sum_{i=0}^{0} i = 0 = 0\frac{0+1}{2}$.
- $n = k+1$, $k \in \mathbb{N}$:

$$\sum_{i=0}^{n} i = \sum_{i=0}^{k+1} i = (k+1) + \sum_{i=0}^{k} i =$$

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

$$(k+1)\left(1 + \frac{k}{2}\right) = (k+1)\frac{k+2}{2}.$$

Inductively defined sets

Inductively defined sets

The natural numbers:

$$\frac{}{\textsf{zero} \in \mathbb{N}} \qquad \frac{n \in \mathbb{N}}{\textsf{suc}\ n \in \mathbb{N}}$$

Compare:

```
data Nat = Zero | Suc Nat
```

Inductively defined sets

Booleans:

$$\frac{}{\mathsf{true} \in Bool} \qquad \frac{}{\mathsf{false} \in Bool}$$

Compare:

```
data Bool = True | False
```

Inductively defined sets

Finite lists:

$$\frac{}{\mathsf{nil} \in \mathit{List}\ A} \qquad \frac{x \in A \qquad xs \in \mathit{List}\ A}{\mathsf{cons}\ x\ xs \in \mathit{List}\ A}$$

Compare:

```
data List a = Nil | Cons a (List a)
```

Which of the following expressions are lists of natural numbers (members of $List\,\mathbb{N}$)?

1. nil.
2. cons nil 5.
3. cons 5 nil.
4. $\mathbf{let}\ xs =$ cons $5\ xs\ \mathbf{in}\ xs$.

Lists

Alternative notation for lists:

- $[\,]$ instead of nil.
- $x : xs$ instead of cons $x\ xs$.
- $[1, 2, 3]$ instead of cons 1 (cons 2 (cons 3 nil)).

Recursive functions

Recursive functions

An example:

$$\begin{aligned} &length \in List\ A \to \mathbb{N} \\ &length\ \mathsf{nil} &&= \mathsf{zero} \\ &length\ (\mathsf{cons}\ x\ xs) &&= \mathsf{suc}\ (length\ xs) \end{aligned}$$

Recursive functions

Not well-defined:

$$
\begin{aligned}
&bad \in List\ A \to \mathbb{N} \\
&bad\ \mathsf{nil} &&= \mathsf{zero} \\
&bad\ (\mathsf{cons}\ x\ xs) &&= bad\ (\mathsf{cons}\ x\ xs)
\end{aligned}
$$

Recursive functions

Another example:

$$
\begin{aligned}
& f \in List\ A \rightarrow List\ A \\
& f\ xs = g\ xs\ \mathsf{nil} \\
& g \in List\ A \rightarrow List\ A \rightarrow List\ A \\
& g\ \mathsf{nil} \qquad\qquad\quad ys = ys \\
& g\ (\mathsf{cons}\ x\ xs)\ ys = g\ xs\ (\mathsf{cons}\ x\ ys)
\end{aligned}
$$

What is the result of $f\,[1, 2, 3]$?

1. $[1, 2, 3]$.
2. $[1, 3, 2]$.
3. $[2, 1, 3]$.
4. $[2, 3, 1]$.
5. $[3, 1, 2]$.
6. $[3, 2, 1]$.

Recursive functions

$$reverse \in List\ A \to List\ A$$
$$reverse\ xs = rev\text{-}app\ xs\ \mathsf{nil}$$

$$\begin{aligned}
&rev\text{-}app \in List\ A \to List\ A \to List\ A \\
&rev\text{-}app\ \mathsf{nil}\ \qquad\qquad\quad ys = ys \\
&rev\text{-}app\ (\mathsf{cons}\ x\ xs)\ ys = rev\text{-}app\ xs\ (\mathsf{cons}\ x\ ys)
\end{aligned}$$

Mutual induction

Mutual induction

- Two mutually defined functions:

$$odd, even \in \mathbb{N} \to Bool$$

$$\begin{aligned} odd\ \mathsf{zero} &= \mathsf{false} \\ odd\ (\mathsf{suc}\ n) &= even\ n \end{aligned}$$

$$\begin{aligned} even\ \mathsf{zero} &= \mathsf{true} \\ even\ (\mathsf{suc}\ n) &= odd\ n \end{aligned}$$

- Another function:

$$\begin{aligned} & odd' \in \mathbb{N} \to Bool \\ & odd'\ \mathsf{zero} = \mathsf{false} \\ & odd'\ (\mathsf{suc}\ n) = not\ (odd'\ n) \end{aligned}$$

- Can we prove $\forall n \in \mathbb{N}.\ odd\ n = odd'\ n$?

Mutual induction

First attempt:

- Let us use mathematical induction.
- Inductive hypothesis:

$$P\,n := odd\,n = odd'\ n$$

- Base case (P zero):

$$\begin{aligned} & odd\,\textsf{zero} & = \\ & \textsf{false} & = \\ & odd'\ \textsf{zero} \end{aligned}$$

Mutual induction

Step case ($\forall n \in \mathbb{N}.\ P\ n \Rightarrow P\ (\mathsf{suc}\ n)$):

- Given $n \in \mathbb{N}$, let us assume $odd\ n = odd'\ n$:

$$\begin{aligned} odd\ (\mathsf{suc}\ n) &= \\ even\ n &= \{???\} \\ not\ (odd'\ n) &= \\ odd'\ (\mathsf{suc}\ n). \end{aligned}$$

- Let us generalise the inductive hypothesis:

$$\begin{aligned} P\ n := odd\ n = odd'\ n \wedge \\ even\ n = not\ (odd'\ n) \end{aligned}$$

Mutual induction

Base case (P zero):

- First part:

$$\begin{aligned} &odd\ \mathsf{zero} &= \\ &\mathsf{false} &= \\ &odd'\ \mathsf{zero} \end{aligned}$$

- Second part:

$$\begin{aligned} &even\ \mathsf{zero} &= \\ &\mathsf{true} &= \\ &not\ \mathsf{false} &= \\ &not\ (odd'\ \mathsf{zero}) \end{aligned}$$

Mutual induction

Step case ($\forall n \in \mathbb{N}.\ P\ n \Rightarrow P\,(\mathsf{suc}\ n)$):

- Given $n \in \mathbb{N}$, let us assume $odd\ n = odd'\ n$ and $even\ n = not\ (odd'\ n)$.
- First part:

$$
\begin{aligned}
&odd\ (\mathsf{suc}\ n) &&= \\
&even\ n &&= \{\text{By the second IH.}\} \\
&not\ (odd'\ n) &&= \\
&odd'\ (\mathsf{suc}\ n)
\end{aligned}
$$

Mutual induction

Step case ($\forall n \in \mathbb{N}.\ P\ n \Rightarrow P\ (\mathsf{suc}\ n)$):

- Given $n \in \mathbb{N}$, let us assume $odd\ n = odd'\ n$ and $even\ n = not\ (odd'\ n)$.
- Second part:

$$
\begin{aligned}
& even\ (\mathsf{suc}\ n) && = \\
& odd\ n && = \{\text{By the first IH.}\} \\
& odd'\ n && = \\
& not\ (not\ (odd'\ n)) && = \\
& not\ (odd'\ (\mathsf{suc}\ n))
\end{aligned}
$$

Discuss how you would prove $\forall n \in \mathbb{N}.\ even\ n = nots\ n\ \mathsf{true}$.

$$
\begin{aligned}
&nots \in \mathbb{N} \to Bool \to Bool \\
&nots\ \mathsf{zero}\qquad\ b = b \\
&nots\ (\mathsf{suc}\ n)\ b = nots\ n\ (not\ b)
\end{aligned}
$$

$$odd, even \in \mathbb{N} \to Bool$$

$$
\begin{aligned}
&odd\ \mathsf{zero} &&= \mathsf{false} \\
&odd\ (\mathsf{suc}\ n) &&= even\ n
\end{aligned}
$$

$$
\begin{aligned}
&even\ \mathsf{zero} &&= \mathsf{true} \\
&even\ (\mathsf{suc}\ n) &&= odd\ n
\end{aligned}
$$

Today

- Proofs.
- Proofs by induction.
- Inductively defined sets.
- Recursive functions.

Next lecture

- Structural induction.
- Some concepts from automata theory.

- Deadline for the next quiz: 2019-01-25, 15:00.
- Deadline for the following quiz: 2019-01-28, **17:00**.