Finite Automata Theory and Formal Languages TMV027/DIT321– LP4 2017

Lecture 3 Ana Bove

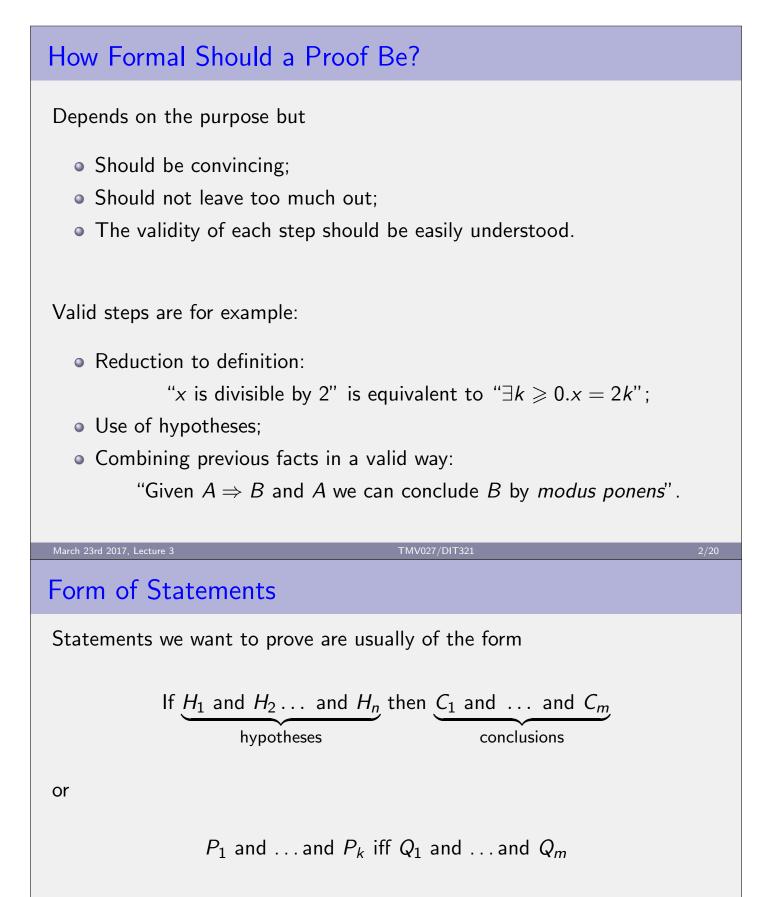
March 23rd 2017

Overview of today's lecture:

- Formal proofs;
- Simple/strong induction;
- Mutual induction;
- Inductively defined sets;
- Recursively defined functions.

Recap: Logic, Sets, Relations, Functions

- Propositions, truth values, connectives, predicates, quantifiers;
- Sets, how to define them, membership, operations on sets, equality, laws;
- Relations, properties (reflexive, symmetric, antisymmetric, transitive, equivalence), partial vs total order, partitions, equivalence class, quotient;
- Functions, domain, codomain, image, partial vs total, injective, surjective, bijective, inverse, composition, restriction.



for $n \ge 0$; $m, k \ge 1$.

Note: Observe that one proves the *conclusion* assuming the validity of the *hypotheses*!

Example: We can easily prove "if 0 = 1 then 4 = 2.000".

TMV027/DIT32

Different Kinds of Proofs

Proofs by Contradiction

If H then C

is logically equivalent to

H and not *C* implies "the impossible" (bottom, \perp).

Example: If $x \neq 0$ then $x^2 \neq 0$ vs. $x \neq 0 \land x^2 = 0 \Rightarrow \bot$

Proofs by Contrapositive

"If H then C" is logically equivalent to "If not C then not H". See both truth tables!

Proofs by Counterexample

We find an example that "breaks" what we want to prove.

Example: All Natural numbers are odd.

March 23rd 2017, Lecture 3

TMV027/DIT321

Proving a Property over the Natural Numbers

How to prove an statement over *all* the Natural numbers?

Example: $\forall n \in \mathbb{N}$. $1 + 2 + 3 + ... + n = \frac{n * (n + 1)}{2}$.

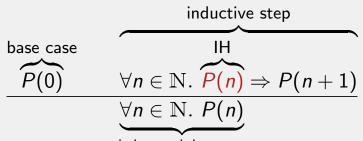
First we need to look at what the Natural numbers are ...

They are an *inductively defined set* defined by the following *two* rules:

$$\frac{n \in \mathbb{N}}{0 \in \mathbb{N}} \qquad \qquad \frac{n \in \mathbb{N}}{n+1 \in \mathbb{N}}$$

(More on inductively defined sets on page 16.)

Mathematical/Simple Induction



statement to prove

More generally:

$$\frac{P(i), P(i+1), \dots, P(j) \quad \forall n \in \mathbb{N}. \ j \leq n \Rightarrow (\overbrace{P(n)}^{\text{IH}} \Rightarrow P(n+1))}{\forall n \in \mathbb{N}. \ i \leq n \Rightarrow P(n)}$$

 $IH \equiv inductive hypothesis$

March 23rd 2017, Lecture 3

TMV027/DI

Example: Proof by Induction

Proposition: Let
$$f(0) = 0$$

 $f(n+1) = f(n) + n + 1$.
Then, $\forall n \in \mathbb{N}$. $f(n) = \frac{n * (n+1)}{2}$.

Proof: By mathematical induction on n.

Let P(n) be $f(n) = \frac{n * (n + 1)}{2}$.

Base case: We prove that P(0) holds.

Inductive step: We prove that if P(n) holds (our IH) for a given $0 \le n$, then P(n+1) also holds.

Closure: Now we have established that for all n, P(n) is true!

In particular, P(0), P(1), P(2), ..., P(15), ... hold.

FMV027/DIT321

Course-of-Values/Strong Induction

Variant of mathematical induction. inductive step base case IH $\forall n \in \mathbb{N}. \ (\forall m \in \mathbb{N}. \ 0 \leq m \leq n \Rightarrow P(m)) \Rightarrow P(n+1)$ P(0) $\forall n \in \mathbb{N}. P(n)$ statement to prove Or more generally: $P(i), P(i+1), \ldots, P(j)$ $\forall n \in \mathbb{N}. \ i < n \Rightarrow (\forall m. \ i \leq m < n \Rightarrow P(m)) \Rightarrow P(n)$ $\forall n \in \mathbb{N}. \ i \leq n \Rightarrow P(n)$ Here we might have several inductive hypotheses P(m)! March 23rd 2017, Lecture 3 Example: Proof by Induction **Proposition:** If $n \ge 8$ then n can be written as a sum of 3's and 5's. **Proof:** By course-of-values induction on n. Let P(n) be "*n* can be written as a sum of 3's and 5's". Base cases: P(8), P(9) and P(10) hold. Inductive step: We shall prove that if $P(8), P(9), P(10), \ldots, P(n)$ hold for $n \ge 10$ (our IH) then P(n+1) holds. Observe that if $n \ge 10$ then $n \ge n + 1 - 3 \ge 8$. Hence by inductive hypothesis P(n + 1 - 3) holds. By adding an extra 3 then P(n+1) holds as well.

Example: All Horses have the Same Colour



Example: Proof by Induction

Proposition: All horses have the same colour.

Proof: By mathematical induction on n.

Let P(n) be "in any set of *n* horses they all have the same colour".

Base cases: P(0) is not interesting in this example. P(1) is clearly true.

Inductive step: Let us show that P(n) (our IH) implies P(n + 1). Let $h_1, h_2, \ldots, h_n, h_{n+1}$ be a set of n + 1 horses. Take h_1, h_2, \ldots, h_n . By IH they all have the same colour. Take now $h_2, h_3, \ldots, h_n, h_{n+1}$. Again, by IH they all have the same colour. Hence, by transitivity, all horses $h_1, h_2, \ldots, h_n, h_{n+1}$ must have the same colour.

Closure: $\forall n$. all *n* horses in the set have the same colour.

Example: What Went Wrong???



Mutual Induction

Sometimes we cannot prove a single statement P(n) but rather a group of statements $P_1(n), P_2(n), \ldots, P_k(n)$ simultaneously by induction on n.

This is very common in automata theory where we need an statement for each of the states of the automata.

Example: Proof by Mutual Induction

Let $f, g, h : \mathbb{N} \to \{0, 1\}$ be as follows:

$$\begin{array}{ll} f(0) = 0 & g(0) = 1 & h(0) = 0 \\ f(n+1) = g(n) & g(n+1) = f(n) & h(n+1) = 1 - h(n) \end{array}$$

Proposition: $\forall n$. h(n) = f(n).

Proof: If P(n) is "h(n) = f(n)" it does not seem possible to prove $P(n) \Rightarrow P(n+1)$ directly.

We strengthen P(n) to P'(n): Let P'(n) be " $h(n) = f(n) \wedge h(n) = 1 - g(n)$ ".

By mathematical induction on n.

We prove $P'(0) : h(0) = f(0) \land h(0) = 1 - g(0)$.

Then we prove that $P'(n) \Rightarrow P'(n+1)$.

Since by induction $\forall n. P'(n)$ is true then $\forall n. P(n)$ is true.

March 23rd 2017, Lecture 3

TMV027/DIT321

Recursive Data Types

What are (the data types of) Natural numbers, lists, trees, ... ?

This is how you would defined them in Haskell:

```
data Nat = Zero | Succ Nat
```

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

```
data BTree a = Leaf a | Node a (BTree a) (BTree a)
```

Observe the similarity between the definition of Nat above and the rules in slide 5...

Inductively Defined Sets

Natural Numbers:

Base case: 0 is a Natural number; Inductive step: If n is a Natural number then n + 1 is a Natural number; Closure: There is no other way to construct Natural numbers.

Finite Lists:

Base case: [] is the empty list over any set A; Inductive step: If $x \in A$ and xs is a list over A then x : xs is a list over A; Closure: There is no other way to construct lists.

Finitely Branching Trees:

Base case: If $x \in A$ then (x) is a tree over any set A; Inductive step: If $x \in A$ and t_1, \ldots, t_k are tree over the set A, then (x, t_1, \ldots, t_k) is a tree over A;

Closure: There is no other way to construct trees.

÷

March 23rd 2017, Lecture 3

FMV027/DIT321

16/20

Inductively Defined Sets (Cont.)

To define a set S by induction we need to specify:

```
Base cases: e_1, \ldots, e_m \in S;
```

Inductive steps: Given $s_1, \ldots, s_{n_i} \in S$, then $c_1[s_1, \ldots, s_{n_1}], \ldots, c_k[s_1, \ldots, s_{n_k}] \in S$;

Closure: There is no other way to construct elements in S. (We will usually omit this part.)

Example: The set of simple Boolean expressions is defined as:

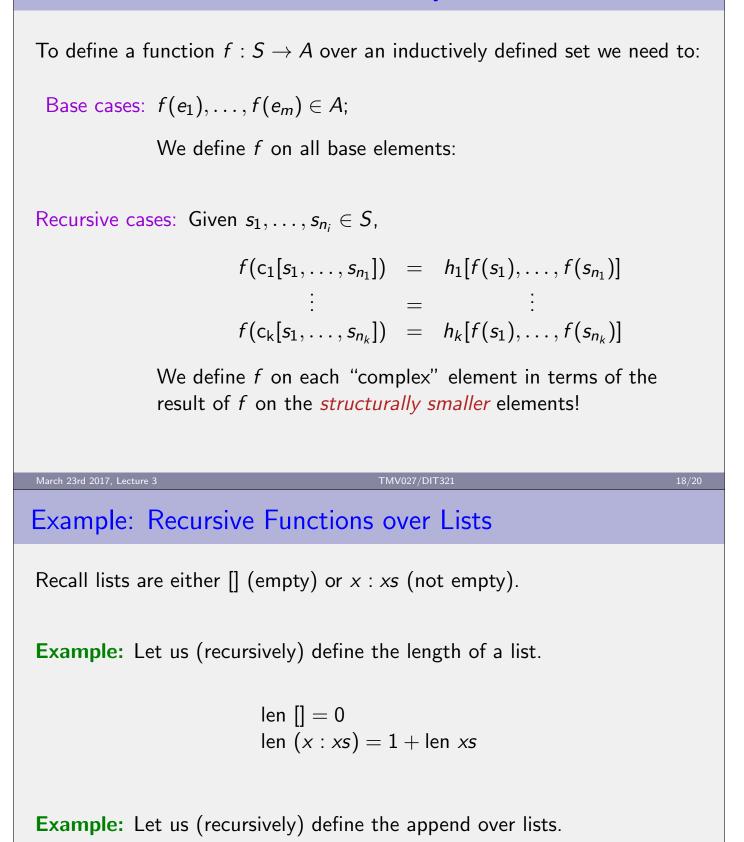
Base cases: true and false are Boolean expressions;

Inductive steps: if *a* and *b* are Boolean expressions then

(a) not a a and b a or b

are also Boolean expressions.

Recursive Functions over Inductively Defined Sets



$$[] ++ ys = ys (x : xs) ++ ys = x : (xs ++ ys)$$

TMV027/DIT32

Chapter 5 in the *Mathematics for Computer Science* book and section 1.2 in the main book:

- Structural induction;
- Concepts of automata theory.

See even Koen Claessen's notes on structural induction (see course web page on Literature).

DO NOT MISS THIS LECTURE!!!

March 23rd 2017, Lecture 3

MV027/DIT321

20/20