Functional Programming

WS 2010/11

Christian Sternagel (VO)
Friedrich Neurauter (PS) Ulrich Kastlunger (PS)

Computational Logic
Institute of Computer Science
University of Innsbruck

November 17, 2010

Today's Topics

- Mathematical Induction
- Induction Over Lists
- Structural Induction

Mathematical Induction

When to use Mathematical Induction?

- prove that some property P holds for all natural numbers
- more formally, prove:

$$\forall n.\, P(n) \qquad (\text{where } n \in \mathbb{N})$$

When to use Mathematical Induction?

- prove that some property P holds for all natural numbers
- more formally, prove:

$$\forall n.\, P(n) \qquad (\text{where } n \in \mathbb{N})$$

How is it Applied?

- mathematical induction consists of two steps:
- first prove base case

$$P(0)$$

- then step case

$$\forall k.\, (P(k) \longrightarrow P(k+1))$$

When to use Mathematical Induction?

- prove that some property P holds for all natural numbers
- more formally, prove:

$$\forall n.\, P(n) \qquad \text{(where } n \in \mathbb{N}\text{)}$$

How is it Applied?

- mathematical induction consists of two steps:
- first prove base case

$$P(0)$$

- then step case

$$\forall k.\, (P(k) \longrightarrow P(k+1))$$

When to use Mathematical Induction?

- prove that some property P holds for all natural numbers
- more formally, prove:

$$\forall n.\, P(n) \qquad \text{(where } n \in \mathbb{N}\text{)}$$

How is it Applied?

- mathematical induction consists of two steps:
- first prove base case

$$P(0)$$

show property for 0

- then step case

$$\forall k.\, (P(k) \longrightarrow P(k+1))$$

When to use Mathematical Induction?

- prove that some property P holds for all natural numbers
- more formally, prove:

$$\forall n.\, P(n) \qquad (\text{where } n \in \mathbb{N})$$

How is it Applied?

- mathematical induction consists of two steps:
- first prove base case

$$P(0)$$

- then step case

$$\forall k.\, (P(k) \longrightarrow P(k+1))$$

When to use Mathematical Induction?

- prove that some property P holds for all natural numbers
- more formally, prove:

$$\forall n.\, P(n) \qquad \text{(where } n \in \mathbb{N}\text{)}$$

How is it Applied?

- mathematical induction consists of two steps:
- first prove base case

$$P(0)$$

- then step case

$$\forall k.\, (P(k) \longrightarrow P(k+1))$$

assume $P(k)$ (induction hypothesis), show $P(k+1)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k + 1$
- and want to show P for every natural number ($\forall n.\, P(n)$)

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k + 1$
- and want to show P for every natural number $(\forall n. P(n))$

Example - $P(3)$

- have $P(0)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k+1$
- and want to show P for every natural number ($\forall n.\, P(n)$)

Example - $P(3)$

- have $P(0)$
- and $P(0) \longrightarrow P(1)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k + 1$
- and want to show P for every natural number ($\forall n. P(n)$)

Example - $P(3)$

- have $P(0)$
- and $P(0) \longrightarrow P(1)$
- thus $P(1)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k+1$
- and want to show P for every natural number $(\forall n.\, P(n))$

Example - $P(3)$

- have $P(0)$
- and $P(0) \longrightarrow P(1)$
- thus $P(1)$
- with $P(1) \longrightarrow P(2)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k+1$
- and want to show P for every natural number ($\forall n.\, P(n)$)

Example - $P(3)$

- have $P(0)$
- and $P(0) \longrightarrow P(1)$
- thus $P(1)$
- with $P(1) \longrightarrow P(2)$
- have $P(2)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k + 1$
- and want to show P for every natural number $(\forall n.\, P(n))$

Example - $P(3)$

- have $P(0)$
- and $P(0) \longrightarrow P(1)$
- thus $P(1)$
- with $P(1) \longrightarrow P(2)$
- have $P(2)$
- with $P(2) \longrightarrow P(3)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k + 1$
- and want to show P for every natural number $(\forall n.\, P(n))$

Example - $P(3)$

- have $P(0)$
- and $P(0) \longrightarrow P(1)$
- thus $P(1)$
- with $P(1) \longrightarrow P(2)$
- have $P(2)$
- with $P(2) \longrightarrow P(3)$
- have $P(3)$

Why does this Work?

- we have two ingredients:
  1. P is true for 0
  2. if P is true for arbitrary k it is also true for $k+1$
- and want to show P for every natural number $(\forall n.\, P(n))$

Example - $P(3)$

- have $P(0)$
- and $P(0) \longrightarrow P(1)$
- thus $P(1)$
- with $P(1) \longrightarrow P(2)$
- have $P(2)$
- with $P(2) \longrightarrow P(3)$
- have $P(3)$

Idea

- intuitively we can reach arbitrary n
- such that $P(n)$
- hence, $\forall n.\, P(n)$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

What is a “Property”?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß’s Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0)$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 \qquad\qquad\qquad)$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 \qquad\qquad)$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$

What is a “Property”?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß’s Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$
- step case: $P(k) \rightarrow P(k+1)$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$
- step case: $P(k) \rightarrow P(k + 1)$
  IH: $P(k) = (1 + 2 + \cdots + k = \frac{k(k+1)}{2})$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$
- step case: $P(k) \rightarrow P(k+1)$
  IH: $P(k) = (1 + 2 + \cdots + k = \frac{k(k+1)}{2})$
  show: $P(k+1)$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$
- step case: $P(k) \rightarrow P(k+1)$
  IH: $P(k) = (1 + 2 + \cdots + k = \frac{k(k+1)}{2})$
  show: $P(k+1)$

$$1 + 2 + \cdots + (k+1)$$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$
- step case: $P(k) \rightarrow P(k+1)$
  IH: $P(k) = (1 + 2 + \cdots + k = \frac{k(k+1)}{2})$
  show: $P(k+1)$

$$1 + 2 + \cdots + (k+1) = (1 + 2 + \cdots + k) + (k+1)$$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$
- step case: $P(k) \rightarrow P(k+1)$
  IH: $P(k) = (1 + 2 + \cdots + k = \frac{k(k+1)}{2})$
  show: $P(k+1)$

$$\begin{aligned} 1 + 2 + \cdots + (k+1) &= (1 + 2 + \cdots + k) + (k+1) \\ &\overset{\text{IH}}{=} \frac{k(k+1)}{2} + (k+1) \end{aligned}$$

What is a "Property"?

- anything that depends on some input and is either true or false
- i.e., some function `p :: a -> Bool`

Example - Gauß's Formula

- $P(x) = (1 + 2 + \cdots + x = \frac{x(x+1)}{2})$
- base case: $P(0) = (1 + 2 + \cdots + 0 = 0 = \frac{0(0+1)}{2})$
- step case: $P(k) \rightarrow P(k+1)$
  IH: $P(k) = (1 + 2 + \cdots + k = \frac{k(k+1)}{2})$
  show: $P(k+1)$

$$\begin{aligned} 1 + 2 + \cdots + (k+1) &= (1 + 2 + \cdots + k) + (k+1) \\ &\overset{\text{IH}}{=} \frac{k(k+1)}{2} + (k+1) \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

Remark

- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

Remark

- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \land \forall k \geq m. (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m. P(n)$$

Remark

- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \land \forall k \geq m.\, (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\, P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls

Remark

- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \land \forall k \geq m.\, (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\, P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls

Remark

- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \wedge \forall k \geq m.\,(P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\,P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls

Remark

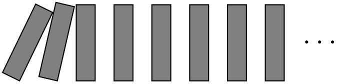
- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \wedge \forall k \geq m.\, (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\, P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls



Remark

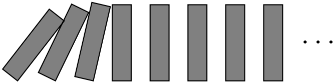
- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \wedge \forall k \geq m.\, (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\, P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls



Remark

- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \wedge \forall k \geq m.\, (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\, P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls

Remark

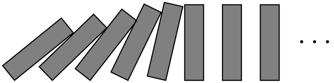
- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \land \forall k \geq m.\, (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\, P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls



Remark

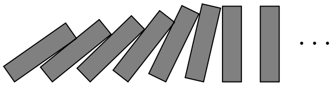
- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \wedge \forall k \geq m. (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m. P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls



Remark

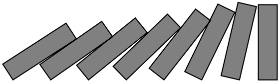
- of course, the base case may be changed
- e.g., if base case $P(1)$, property holds for all $n \geq 1$

General Induction Principle

$$(P(m) \wedge \forall k \geq m.\, (P(k) \longrightarrow P(k+1))) \longrightarrow \forall n \geq m.\, P(n)$$

Domino Effect

- first domino will fall
- if a domino falls also its right neighbor falls



$\ldots$

Induction Over Lists

Recall

- type: `data [a] = [] | (:) a [a]`

Recall

- type: `data [a] = [] | (:) a [a]`

Notes

- lists are recursive structures
- base case: `[]`
- step case: x `:` xs

Induction Principle for Lists - Informally

- to show $P(xs)$ for all lists xs
- show base case: $P(\texttt{[]})$
- show step case: $P(xs) \longrightarrow P(x : xs)$ for arbitrary x and xs

Induction Principle for Lists - Informally

- to show $P(xs)$ for all lists xs
- show base case: $P(\texttt{[]})$
- show step case: $P(xs) \longrightarrow P(x : xs)$ for arbitrary x and xs

Induction Principle for Lists - Formally

$$(P(\texttt{[]}) \land \forall x. \forall xs. (P(xs) \longrightarrow P(x : xs)))$$
$$\longrightarrow \forall ls. P(ls)$$

Induction Principle for Lists - Informally

- to show $P(xs)$ for all lists xs
- show base case: $P(\texttt{[]})$
- show step case: $P(xs) \longrightarrow P(x \mathbin{\texttt{:}} xs)$ for arbitrary x and xs

Induction Principle for Lists - Formally

$$(P(\texttt{[]}) \land \forall x.\, \forall xs.\, (P(xs) \longrightarrow P(x \mathbin{\texttt{:}} xs))) \\ \longrightarrow \forall ls.\, P(ls)$$

Remark

- for lists, P can be seen as function `p :: [a] -> Bool`

Exercise - Right Identity for List Append

- definition

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

Exercise - Right Identity for List Append

- definition

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

- lemma: [] is a right identity of ++, i.e.,

$$xs \mathbin{++} [] = xs$$

Exercise - Associativity of Append

- recall

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

Exercise - Associativity of Append

- recall

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

- and '*xs* ++ [] = *xs*' for all lists *xs*

Exercise - Associativity of Append

- recall

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

- and 'xs ++ [] $= xs$' for all lists xs
- lemma: ++ is associative, i.e.,

$$xs \mathbin{++} (ys \mathbin{++} zs) = (xs \mathbin{++} ys) \mathbin{++} zs$$

Exercise - Length and Append

Exercise - Length and Append

- definition

```
length []       =  0
length (_:xs)   =  1 + length xs
```

Exercise - Length and Append

- definition

```
length []       =  0
length (_:xs)   =  1 + length xs
```

- lemma: sum of lengths is length of combined list, i.e.,

$$\texttt{length}\ xs + \texttt{length}\ ys = \texttt{length}\ (xs\ \texttt{++}\ ys)$$

Structural Induction

Example - Terms

```
data Term = Var String
          | Lab String Term
          | App Term Term
```

Example - Terms

```
data Term = Var String
          | Lab String Term
          | App Term Term
```

General Structures - Induction Principle

- for every non-recursive constructor, show base case
  - base case: $P(\texttt{Var x})$
- for every recursive constructor, show step case
  - step case 1: $P(\texttt{t}) \longrightarrow P(\texttt{Lab x t})$
  - step case 2: $P(\texttt{s}) \wedge P(\texttt{t}) \longrightarrow P(\texttt{App s t})$

Example - Binary Trees

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

Example - Binary Trees

```haskell
data BTree a = Empty
             | Node a (BTree a) (BTree a)
```

Induction Principle for Binary Trees

$$(P(\texttt{Empty}) \land \forall x. \forall l. \forall r. (P(l) \land P(r) \longrightarrow P(\texttt{Node}\ x\ l\ r))) \\ \longrightarrow \forall t. P(t)$$

Exercise - Perfect Binary Trees

- a binary tree is perfect if all leaf nodes have same depth

```
perfect Empty          = True
perfect (Node x l r) =
  height l == height r && perfect l && perfect r

height Empty          = 0
height (Node _ l r) =
  max (height l) (height r) + 1

size Empty          = 0
size (Node _ l r) = size l + size r + 1
```

- lemma: a perfect binary tree t of height n has exactly $2^n - 1$ nodes, i.e.,

$$P(t) = (\texttt{perfect}\ t \longrightarrow \texttt{size}\ t = 2^{\texttt{height}\ t} - 1)$$

Exercises (for November 26th)

1. Prepare for the 1st test!
2. Prove the following equation by induction

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

3. Prove $\texttt{rev}\ (xs \mathbin{+\!+} ys) = \texttt{rev}\ ys \mathbin{+\!+} \texttt{rev}\ xs$ for

```
rev []      =  []
rev (x:xs)  =  rev xs ++ [x]
```

using the equations

$$xs \mathbin{+\!+} [] = xs \qquad (\star)$$

$$(xs \mathbin{+\!+} ys) \mathbin{+\!+} zs = xs \mathbin{+\!+} (ys \mathbin{+\!+} zs) \qquad (\star\star)$$