

Foundations of Computer Science

19. Mathematical induction

Summary: This lecture introduces mathematical induction as a technique for proving the equivalence of two functions, or for proving properties of functions.

Reference: Thompson Chapter 8

Recursion and induction

- Mathematical induction is a proof technique closely related to recursion
 - remember the distinction between scientific induction and mathematical induction
 - scientific induction
 - \cong inductive reasoning
 - \cong program validation
 - \cong testing
 - “reasoning from the particular to the general”
 - mathematical induction
 - \cong deductive reasoning
 - \cong program verification
 - “reasoning from the general to the particular”
- Remember how recursive functions are executed

```
(^) :: Int -> Int -> Int
-- pre: n >= 0
-- x^n returns the nth power of x
x ^ n | n == 0 = 1
      | n > 0  = x * x ^ (n - 1)
```

```
Hugs> 7 ^ 3
=> 7 * 7 ^ 2
=> 7 * (7 * 7 ^ 1)
=> 7 * (7 * (7 * 7 ^ 0))
=> 7 * (7 * (7 * 1))
```

343

- several steps with the recursive equation, followed by one step with the base case

Proofs of equivalence

- Consider two definitions of `sumnats`

```
sumnats, sumnats' :: Int -> Int
-- pre: n >= 0
-- sumnats n returns the sum of the
-- first n positive integers
sumnats 0 = 0
sumnats n = sumnats (n - 1) + n

-- pre: n >= 0
-- sumnats' n returns the sum of the
-- first n positive integers
sumnats' n = n * (n + 1) `div` 2
```

- How do we know whether these are equivalent?
 - i.e. how do we know if they **always** give the same answer?
- State a theorem and prove it

```
sumnats n = sumnats' n
```

– or, more formally

“for all n that satisfy the pre-condition:
 $\text{sumnats } n = \text{sumnats}' n$ ”

Mathematical induction

- A proof using induction over the natural numbers has two parts
 - prove that the theorem is true for 0
 - prove that **if the theorem is true for k** , then it is true for $k + 1$
 - the assumption in bold here is known as the **inductive hypothesis**
- The logical consequence of these two parts together is that the theorem is true for all n
 - it is true for 0 by direct proof
 - it is true for 1 **because it is true for 0**
 - it is true for 2 **because it is true for 1**
 - it is true for 3 **because it is true for 2**
 - etc.
 - it is true for $k+1$ **because it is true for k**
- This argument uses the logical principle of *modus ponens*:
 - if A is true
 - and if $A \rightarrow B$ (i.e. A implies B)
 - then B is true
 - this is written formally as $A \wedge (A \rightarrow B) \rightarrow B$
- cf. ladders and dominoes
- By convention, each part of the proof follows a standard format
 - we start with the left-hand side and use a sequence of steps to transform it to the right-hand side
 - **each step must be justified**, either by reference to a particular definition or by appealing to properties of built-in functions

Example: the equivalence of the two sumnats definitions

- Theorem:

$$\text{sumnats } n = \text{sumnats}' n$$

Case 0:

Prove that $\text{sumnats } 0 = \text{sumnats}' 0$

$$\begin{aligned} \text{sumnats } 0 &= 0 && \text{sumnats} \\ &= 0 * (0 + 1) \text{ `div` } 2 && \text{algebra} \\ &= \text{sumnats}' 0 && \text{sumnats}' \end{aligned}$$

Case k+1:

Prove that if $\text{sumnats } k = \text{sumnats}' k$

then $\text{sumnats } (k+1) = \text{sumnats}' (k+1)$

$$\begin{aligned} \text{sumnats } (k+1) &= \text{sumnats } (k+1-1) + k + 1 && \text{sumnats} \\ &= \text{sumnats } k + k + 1 && \text{algebra} \\ &= \text{sumnats}' k + k + 1 && \text{ind. hyp.} \\ &= k * (k+1) \text{ `div` } 2 + k + 1 && \text{sumnats}' \\ &= (k * (k+1) + 2 * (k+1)) \text{ `div` } 2 && \text{algebra} \\ &= (k+2) * (k+1) \text{ `div` } 2 && \text{algebra} \\ &= (k+1) * (k+2) \text{ `div` } 2 && \text{algebra} \\ &= (k+1) * (k+1+1) \text{ `div` } 2 && \text{algebra} \\ &= \text{sumnats}' (k+1) && \text{sumnats}' \end{aligned}$$

Proof of a theorem over \wedge

- Consider the recursive definition of \wedge

```
( $\wedge$ ) :: Int -> Int -> Int
-- pre: n >= 0
--  $x^n$  returns the nth power of x
 $x \wedge n$  | n == 0 = 1
          | n > 0  = x *  $x \wedge (n - 1)$ 
```

- Prove the theorem

$$x \wedge (n + m) = x \wedge n * x \wedge m$$

- Which number do we induce over?
 - either n or m , because the function recurses over its second argument

Proof of a theorem over \wedge contd.

Case 0:

Prove that $x \wedge (0 + m) = x \wedge 0 * x \wedge m$

$$\begin{aligned}x \wedge (0 + m) &= x \wedge m && \text{id of } + \\ &= 1 * x \wedge m && \text{id of } * \\ &= x \wedge 0 * x \wedge m && \wedge\end{aligned}$$

Case $k+1$:

Prove that if $x \wedge (k + m) = x \wedge k * x \wedge m$

then $x \wedge (k + 1 + m) = x \wedge (k + 1) * x \wedge m$

$$\begin{aligned}x \wedge (k + 1 + m) &= x \wedge (k + m + 1) && \text{assoc. of } + \\ &= x * x \wedge (k + m) && \wedge \\ &= x * (x \wedge k * x \wedge m) && \text{ind. hyp.} \\ &= (x * x \wedge k) * x \wedge m && \text{assoc. of } * \\ &= x \wedge (k + 1) * x \wedge m && \wedge\end{aligned}$$

- Note that in **searching for** a proof, you might work “from both ends to the middle”, but the final proof **must** be presented from left-to-right

The two aspects of induction

- There are two important aspects to understanding and using mathematical induction
 - understand how the two parts of a proof work together to prove the theorem for all possible argument values
 - cf. ladders and dominoes
 - understand how to construct a proof
 - proof by two parts
 - state and use the inductive hypothesis
- Don't get these mixed-up!