

50003 - Models of Computation - (Dr Raad) Lecture 5

Oliver Killane

25/10/21

Structural Induction

Structural induction is used for reasoning about collections of objects, which are:

- structured in a well defined way
- finite but can be arbitrarily large and complex

We can use this to reason about:

- natural numbers
- data structures (lists, trees, etc)
- programs (can be large, but are finite)
- derivations of assertions like $E \Downarrow 4$ (finite trees of axioms and rules)

Structural Induction over Natural Numbers

$$\mathbb{N} \in \text{Nat} ::= \text{zero} \mid \text{succ}(\mathbb{N})$$

To prove a property $P(\mathbb{N})$ holds, for every number $N \in \text{Nat}$ by induction on structure \mathbb{N} :

- **Base Case** Prove $P(\text{zero})$
- **Inductive Case** Inductive Case is $P(\text{Succ}(K))$ where $P(K)$ holds

For example, we can prove the property:

$$\text{plus}(\mathbb{N}, \text{zero}) = \mathbb{N}$$

- **Base Case**

Show $\text{plus}(\text{zero}, \text{zero}) = \text{zero}$

$$\begin{array}{lll} (1) & \text{LHS} & = \text{plus}(\text{zero}, \text{zero}) \\ (2) & & = \text{zero} & \text{(By definition of plus)} \\ (3) & & = \text{RHS} & \text{(As Required)} \end{array}$$

- **Inductive Case**

$N = \text{succ}(K)$

Inductive Hypothesis $\text{plus}(K, \text{zero}) = K$

Show $\text{plus}(\text{succ}(K), \text{zero}) = \text{succ}(K)$

$$\begin{array}{lll} (1) & \text{LHS} & = \text{plus}(\text{succ}(K), \text{zero}) \\ (2) & & = \text{succ}(\text{plus}(K, \text{zero})) & \text{(By definition of plus)} \\ (3) & & = \text{succ}(K) & \text{(By Inductive Hypothesis)} \\ (4) & & = \text{RHS} & \text{(As Required)} \end{array}$$

Mathematics induction is a special case of structural induction:

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

In the exam you may use $P(0)$ and $P(K+1)$ rather than $P(\text{zero})$ and $P(\text{succ}(k))$ to save time.

Binary Tree Example

$$bTree \in BinaryTree ::= Node | Branch(bTree, bTree)$$

We can define a function *leaves*:

$$leaves(Node) = 1$$

$$leaves(Branch(T_1, T_2)) = leaves(T_1) + leaves(T_2)$$

Or *branches*:

$$branches(Node) = 0$$

$$branches(Branch(T_1, T_2)) = branches(T_1) + branches(T_2) + 1$$

Exercise

Prove By induction that $leaves(T) = branches(T) + 1$

Induction over SimpleExp

$$E \in SimpleExp ::= n | E + E | E \times E | \dots$$

where $n \in \mathbb{N}$.

Properties of \Downarrow

- **Determinacy**

A simple expression can only evaluate to one answer.

$$E \Downarrow n_1 \wedge E \Downarrow n_2 \rightarrow n_1 = n_2$$

- **Totality**

A simple expression evaluates to at least one answer.

$$\forall E \in SimpleExp. \exists n \in \mathbb{N}. [E \Downarrow n]$$