Scribe: Sigmund Cherem & Oren Kurland Lecturer: Radu Rugina

1 Introduction to Induction

We want to be able to prove that certain properties of programs hold, such as:

- termination
- non-determinism
- equivalence of programs and expressions
- equivalence of semantics

Induction is a technique that allows us to develop two diferent tasks: Define sets inductively and make proofs about properties in those sets. Programs are built using inductive definitions, thus we could use induction to prove those properties.

1.1 Inductive definition of sets

An inductive set has two types of elements:

- Basic elements: elements are by default in the set.
- Inductive elements: set elements defined by modification, combination or alteration of other elements in the set.

Some examples are:

- Natural numbers ($\mathbb{N}$), where the basic element is 0 and the inductive elements are built by adding 1 to an existing natural number.
- Syntactic constructs ($\mathbf{Bexp}, \mathbf{Aexp}, \mathbf{Com}$), where the basic elements are the terminal symbols and the inductive elements are those composed by terminal symbols or other inductive elements.
- Proof trees, where the basic elements are axioms and the inductive elements are trees built by applying semantic rules.
- Sequences of excecution (in small-steps semantics), where the basic element is the sequence of zero steps, and the inductive elements are sequences of excecution.

1.2 Contructing a proof

When contructing a proof by induction, we have to prove two separate things:

- The property should hold for the basic elements (Basis or base case).
- The property should hold for an inductive element if this property holds for the componebts used to build this element (Induction step).

2 Types of Induction

2.1 Standard induction

Suppose we have a property $P(n)$ and we want to prove that

$$\forall n \geqslant 0 \ . \ P(n)$$

The proof will look as follows:

- Base case: $P(0)$ holds.
- Induction step: $\forall n \geqslant 0 \ . \ P(n) \Rightarrow P(n+1)$.

2.2 Course of values induction

The proof will look as follows:

- Base case: $P(0)$ holds.
- Induction step: $\forall n \geqslant 0 \ . \ [(\forall k \leqslant n \ . \ P(k)) \Rightarrow P(n+1)]$.

Both cases can be combined into one formula:

$$\forall n \geqslant 0 \ \ [(\forall k < n \ . \ P(k)) \Rightarrow P(n)]$$

Course of values induction and standard induction are equivalent. To see this we define $P'(n)$ as a new property in terms of $P(n)$, where standard induction for $P'(n)$ means course of values induction for $P(n)$:

$$P'(n) = \forall k \in [0, n] \ . \ P(k)$$

2.3 Structural Induction

We will use structural induction in order to prove properties about syntactic constructs. For instance, we would like to prove that

$$\forall a \in \textbf{Aexp} \ . \ P(a)$$

To prove it, we need to recall the following syntax:

$$a ::= n \mid X \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 * a_1$$

Where n and X are basic elements and the rest are inductive defined elements.
A proof should look as follows:

- Base case: $(\forall n \ . \ P(n)) \wedge (\forall x \in Loc \ . \ P(X))$
- Inductive step:

$$\forall a_0, a_1 \in \textbf{Aexp} \ . \ (P(a_0) \wedge P(a_1)) \Rightarrow (P(a_0 + a_1) \wedge P(a_0 - a_1) \wedge P(a_0 * a_1))$$

Example: The evaluation of all arithmetic expressions is deterministic.

$$P(a) \triangleq \forall \sigma, m, m' \ . \ (\langle a, \sigma \rangle \to m \wedge \langle a, \sigma \rangle \to m') \Rightarrow m = m'$$

Proof:

1. Base case:

- $\forall n \ . \ P(n)$
  As a notation, we used the { } as an explanation of the assumption used in each step of the proof.

$$
\begin{array}{rl}
 & \forall \sigma, m, m' \ . \ (\langle n,\, \sigma\rangle \to m \land \langle n,\, \sigma\rangle \to m') \\
\Rightarrow & \quad \left\{ \text{applying axiom } \overline{\langle n,\, \sigma\rangle \to n} \text{ twice} \right\} \\
 & (m = n \land m' = n) \\
\Rightarrow & \quad \{\ m = n = m'\ \} \\
 & m = m'
\end{array}
$$

  Therefore $\boxed{\forall \sigma, m, m' \ . \ (\langle n,\, \sigma\rangle \to m \land \langle n,\, \sigma\rangle \to m') \Rightarrow m = m'}$

- $\forall X \ . \ P(X)$
  In this case the proof looks as follows:

$$
\begin{array}{rl}
 & \forall \sigma, m, m' \ . \ (\langle X,\, \sigma\rangle \to m \land \langle X,\, \sigma\rangle \to m') \\
\Rightarrow & \quad \left\{ \text{applying axiom } \overline{\langle X,\, \sigma\rangle \to \sigma(x)} \text{ twice} \right\} \\
 & (m = \sigma(x) \land m' = \sigma(x)) \\
\Rightarrow & \quad \{\ \text{Since } \sigma(x) \text{ is a function, } m = \sigma(x) = m'\ \} \\
 & m = m'
\end{array}
$$

  Therefore $\boxed{\forall \sigma, m, m' \ . \ (\langle X,\, \sigma\rangle \to m \land \langle X,\, \sigma\rangle \to m') \Rightarrow m = m'}$

2. Induction step: We want to prove that

$$\forall a_0, a_1 \ . \ P(a_0) \land P(a_1) \Rightarrow P(a_0 + a_1)$$

Proof:

$$
\begin{array}{rl}
 & \langle a_0 + a_1,\, \sigma\rangle \to p \land \langle a_0 + a_1,\, \sigma\rangle \to p' \\
\Rightarrow & \left\{ \begin{array}{l} \text{from rule } \dfrac{\langle a_0,\, \sigma\rangle \to m \quad \langle a_1,\, \sigma\rangle \to n}{\langle a_0 + a_1,\, \sigma\rangle \to p} \ p = m + n \\[2ex] \text{and rule } \dfrac{\langle a_0,\, \sigma\rangle \to m' \quad \langle a_1,\, \sigma\rangle \to n'}{\langle a_0 + a_1,\, \sigma\rangle \to p'} \ p' = m' + n' \end{array} \right\} \\
 & \exists m, n \ . \ (\langle a_0,\, \sigma\rangle \to m \ \land \ \langle a_1,\, \sigma\rangle \to n \ \land \ p = n + m) \ \land \\
 & \exists m', n' \ . \ (\langle a_0,\, \sigma\rangle \to m' \ \land \ \langle a_1,\, \sigma\rangle \to n' \ \land \ p' = n' + m') \\
\Rightarrow & \left\{ \begin{array}{l} \text{By induction hypothesis,} \\ ((\langle a_0,\, \sigma\rangle \to m \land \langle a_0,\, \sigma\rangle \to m') \Rightarrow m = m' \text{ by } P(a_0) \\ ((\langle a_1,\, \sigma\rangle \to n \land \langle a_1,\, \sigma\rangle \to n') \Rightarrow n = n' \text{ by } P(a_1) \end{array} \right\} \\
 & p = n + m = p'
\end{array}
$$

Therefore $\boxed{\forall \sigma, p, p' \ . \ (\langle a_0 + a_1,\, \sigma\rangle \to p \land \langle a_0 + a_1,\, \sigma\rangle \to p') \Rightarrow p = p'}$

The proof for the other binary operations ($-$ and $*$) is the same. Therefore we have demostrated that this is true for any **Aexp**:

$$\boxed{\forall a, \sigma, m, m' \ . \ (\langle a,\, \sigma\rangle \to m \land \langle a,\, \sigma\rangle \to m') \Rightarrow m = m'}$$

Another example: The evaluation of all commands is deterministic.

$$P(c) \triangleq \forall \sigma, \sigma', \sigma'' \ . \ (\langle c,\, \sigma\rangle \to \sigma' \land \langle c,\, \sigma\rangle \to \sigma'') \Rightarrow \sigma' = \sigma''$$

We will consider just the hard part and try to prove the property only for the **while** command.

$$c ::= \ldots \mid \textbf{while}\, b\, \textbf{do}\, c$$

We want to prove that

$$\forall c\ .\ P(c) \Rightarrow P(\textbf{while}\, b\, \textbf{do}\, c)$$

We will just analyze the case in which $\langle b, \sigma\rangle \to \textbf{true}$ and we will use the notation $w \equiv \textbf{while}\, b\, \textbf{do}\, c$. We want to prove that $\forall \sigma, \sigma', \sigma''\ .\ (\langle w, \sigma\rangle \to \sigma' \wedge \langle w, \sigma\rangle \to \sigma'') \Rightarrow \sigma' = \sigma''$. Using the semantic rule of the **while** we get:

$$\frac{\langle b, \sigma\rangle \to \textbf{true} \quad \langle c, \sigma\rangle \to \sigma_1 \quad \langle w, \sigma_1\rangle \to \sigma'}{\langle w, \sigma\rangle \to \sigma'} \quad \frac{\langle b, \sigma\rangle \to \textbf{true} \quad \langle c, \sigma\rangle \to \sigma_2 \quad \langle w, \sigma_2\rangle \to \sigma''}{\langle w, \sigma\rangle \to \sigma''}$$

Because $\langle c, \sigma\rangle \to \sigma_1$, $\langle c, \sigma\rangle \to \sigma_2$, and the evaluation of c is deterministic by induction hypothesis, we get $\sigma_1 = \sigma_2$. Using this result and the other premises of the above rules, we get $\langle w, \sigma_1\rangle \to \sigma'$ and $\langle w, \sigma_1\rangle \to \sigma''$. Remember that we want to prove $\sigma' = \sigma''$. Essentially we reduced the original problem to the same problem, so structural induction fails to prove our desired property. We therefore need another kind of induction to help us prove that the execution of commands is deterministic.

2.4 Induction on derivations

We define the relation between proof trees as follows:

$$D_1 \prec D_2 \Leftrightarrow D_1 \text{ is included in } D_2$$

In order to prove that $\forall d\ .\ P(d)$, we use the induction as follows:

- Base case: prove facts about axioms.
- Inductive step: $(\forall d' \prec d\ .\ P(d')) \Rightarrow P(d)$.

We will introduce the following notation:

- $d \Vdash \langle c, \sigma\rangle \to \sigma'$ means that derivation d proves $\langle c, \sigma\rangle \to \sigma'$.
- $\langle c, \sigma\rangle \to \sigma'$ means $\exists e\ .\ e \Vdash \langle c, \sigma\rangle \to \sigma'$.

Example We will now use the induction on derivations to prove that the execution of commands is deterministic. For this, we define the following property:

$$P(d) \triangleq \forall c, \sigma, \sigma', \sigma''\ .\ [(d \Vdash \langle c, \sigma\rangle \to \sigma') \wedge \langle c, \sigma\rangle \to \sigma''] \Rightarrow \sigma' = \sigma''$$

Our inductive hypothesis is that $\forall d' \prec d\ .\ P(d')$. Again, we will focus on the hard part of the proof, which is the case of a **while** command whose condition evaluates to **true**. Let d be a derivation for such a command. We want to prove: $d \Vdash \langle w, \sigma\rangle \to \sigma' \ \wedge\ \langle w, \sigma\rangle \to \sigma'' \ \Rightarrow\ \sigma' = \sigma''$, knowing that $\forall d' \prec d\ .\ P(d')$.

From $d \Vdash \langle w, \sigma\rangle \to \sigma'$, because the condition of w evaluates to **true**, we can conclude that there exist two derivations $d', d'' \prec d$ such that

$$d' \Vdash \langle c, \sigma\rangle \to \sigma_1$$
$$d'' \Vdash \langle w, \sigma_1\rangle \to \sigma'$$

From $\langle c, \sigma\rangle \to \sigma''$ we get that:

$$\langle c, \sigma\rangle \to \sigma_2$$
$$\langle w, \sigma_2\rangle \to \sigma''$$

We can apply our inductive hypothesis $P(d')$ and get the result that $\sigma_1 = \sigma_2$. Then

$$d'' \Vdash \langle w, \sigma_1\rangle \to \sigma'$$
$$\langle w, \sigma_1\rangle \to \sigma''$$

We can apply our inductive hypothesis $P(d'')$ and finally conclude that $\sigma' = \sigma''$.