CS611 Lecture 07 **Induction and Inductive Definitions** 12 September 2003

Scribe: David Martin, Tom Finley Lecturer: Radu Rugina

---

1 Induction on Derivations (contd.)

In the previous lecture, we attempted to prove that commands execute deterministically within IMP. That is:

$$\forall c. \forall \sigma, \sigma', \sigma''. \langle c, \sigma\rangle \to \sigma' \wedge \langle c, \sigma\rangle \to \sigma'' \Rightarrow \sigma' = \sigma''$$

We encountered problems when using structural induction to prove that **while** was deterministic. We discovered that structural induction was not powerful enough to prove this property and we had to use *induction on derivations.*

We first define $\prec$, a binary relation on derivations, by $d \prec d'$ iff d is a subderivation of d', where d and d' are derivations. That is, $d \prec d'$ iff the proof tree of d occurs as a subtree of the proof tree of d'.

Induction on derivation amounts to showing that:

$$\forall d.\ (\forall d' \prec d.\ P(d')) \Rightarrow P(d)$$

That is, we have to prove $P(d)$ assuming that$P$ holds for all subderivations of d.

1.1 Notation

We write $d \Vdash \langle c, \sigma\rangle \to \sigma'$ to indicate that d is a derivation of $\langle c, \sigma\rangle \to \sigma'$.

Since $\langle c, \sigma\rangle \to \sigma' \equiv \exists e.\ e \Vdash \langle c, \sigma\rangle \to \sigma'$, we do not need to explicitly state the existence of e to say that $\langle c, \sigma\rangle \to \sigma'$ is factual.

1.2 Proof of Nondeterminism

We can now state the property of determinism P as follows:

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

The complete proof examines all the possible forms of commands that c may take. That is, we have cases for **skip**, for assignments $X := a$, for sequences $c_0; c_1$, for if commands **if** b **then** c_0 **else** c_1, and for loops **while** b **do** c_0. Here we only present the proof in the case of loops; we leave all of the other cases as an exercise.

Case $c \equiv$ **while** b **do** c_0. We denote this while statement by w. Consider states σ, σ', and σ'' and a derivation d such that $d \Vdash \langle c, \sigma\rangle \to \sigma'$ and $\langle c, \sigma\rangle \to \sigma''$.

We know that $d \Vdash \langle c, \sigma\rangle \to \sigma'$. By inspecting all the semantic rules, there are two rules that matches this conclusion, the rules for while loops. Hence, the derivation d for this is of one of the following forms:

$$\frac{\dfrac{\vdots}{\langle b,\sigma\rangle \to \mathit{false}}}{\langle w,\sigma\rangle \to \sigma} \quad \text{or} \quad \frac{\dfrac{\vdots}{\langle b,\sigma\rangle \to \mathit{true}} \quad \boxed{\dfrac{\vdots}{\langle c_0,\sigma\rangle \to \sigma_0}}^{=d'} \quad \boxed{\dfrac{\vdots}{\langle w,\sigma_0\rangle \to \sigma'}}^{=d''}}{\langle w,\sigma\rangle \to \sigma'}$$

We consider two subcases, depending on whether b evaluated to true or false in state σ.

- Subcase $\langle b, \sigma\rangle \to \mathit{false}$. In this case, d is the left tree, which shows that $\sigma = \sigma'$. Also, since b evaluates to *false*, the derivation for $\langle c, \sigma\rangle \to \sigma''$ will be the same as the left tree above. Hence $\sigma = \sigma'$. Therefore $\sigma' = \sigma''$.

- Subcase $\langle b, \sigma \rangle \to \mathit{true}$. The derivation d for $\langle c, \sigma \rangle \to \sigma'$ is the right tree above.

  We also know that $\langle c, \sigma \rangle \to \sigma''$, which means that there exists a derivation for this relation. By inspecting all the semantic rules, there are two rules that match this conclusion, the rules for while loops. Moreover, since $\langle b, \sigma \rangle \to \mathit{true}$, only one of these rules applies. Hence, the derivation for this relation has the following form:

$$\frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \mathit{true}} \quad \dfrac{\vdots}{\langle c_0, \sigma \rangle \to \sigma_1} \quad \dfrac{\vdots}{\langle w, \sigma_1 \rangle \to \sigma''}}{\langle w, \sigma \rangle \to \sigma''}$$

  This derivation shows that $\langle c_0, \sigma \rangle \to \sigma_1$. From the form of derivation d we know that $d' \Vdash \langle c_0, \sigma \rangle$. Since $d' \prec d$, we can apply the Induction Hypothesis $P(d')$ and get: $\sigma_0 = \sigma_1$.

  The derivation for $\langle c, \sigma \rangle \to \sigma''$ also shows that $\langle w, \sigma_1 \rangle \to \sigma''$. Since $\sigma_0 = \sigma_1$, it means that $\langle w, \sigma_0 \rangle \to \sigma''$. From the form of derivation d we know that $d'' \Vdash \langle c_0, \sigma \rangle$. Since $d'' \prec d$, we can apply the Induction Hypothesis $P(d'')$ and get: $\sigma' = \sigma''$, which completes the proof of this case.

2 Well-founded Induction

All the forms of induction that we have seen so far have had similarities. We will now attempt to formalize this as a more general framework for induction.

Given a set A and a relation $\prec$ on $A \times A$, we say that A *is well-founded with respect to* $\prec$ iff there are no infinite $\prec$-descending chains in A. That is, for every $a \in A$, all of the descending chains starting at a are of the form $a_0 \prec \cdots \prec a_n \prec a$ and $\forall a' \in A \ . \ a' \not\prec a_0$.

In such cases, proof by induction amounts to showing that:

$$\forall a. \ (\forall b \prec a. \ P(b)) \Rightarrow P(a)$$

Note that all previously encountered forms of induction have, in fact, been on well-founded sets:

- Standard induction: $A = \mathbb{N}$; $\prec = \{(m, n) : m + 1 = n\}$.
- Course-of-values induction: $A = \mathbb{N}$; $\prec = \{(m, n) : m < n\}$.
- Structural induction: $A = \mathbf{Aexp}$; $\prec = \{(a_0, a_1) : a_0$ is a subexpresion of $a_1\}$.
- Induction on derivations: $A = \{d : d \Vdash \langle c, \sigma \rangle \to \sigma'\}$; $\prec = \{(d, d') : d$ is a subderivation of $d'\}$.

3 Inductive Definitions

So far we have see different definitions for inductive sets. These definitions include grammars (for syntactic sets) and semantic rules (for proof trees). We would like to have a standard way to define inductive sets.

An *inductive definition for a set* is a collection of rules, which includes axioms and inference rules. An axiom:

$$\overline{x}$$

indicates that the element x is in the set by default. An inference rule:

$$\frac{x_1 \quad \cdots \quad x_n}{x}$$

shows that x is an element of the set, provided that $x_1, \ldots, x_n$ are elements in the set.

3.0.1 Examples

- The set $\mathbb{N}$ is an inductively defined set, whose definition consists of two rules:

$$\overline{0} \qquad \frac{n}{succ(n)}$$

- The set **Aexp** is an inductively defined set, whose definition consists of the following rules:

$$\overline{n} \qquad \overline{X} \qquad \frac{a_0 \quad a_1}{a_0 \oplus a_1}$$

  Essentially, each rule is a production in the grammar.

3.1 Constructing a set from an inductive definition

Given a set of rules which describe an inductive set, we want to formally define what are all the elements of that set. For this, we define a *rule operator* $\hat{R}(X)$ by:

$$\hat{R}(X) = \{y : \exists X' \subseteq X \text{ s.t. } \frac{X'}{y}\}$$

Intuitively, $\hat{R(X)}$ is the set of elements that can be inferred by applying the rules once, starting from a set X of elements. Denote n applications of the $\hat{R}$ operator by $\hat{R}^n$. That is,

$$\hat{R}^n(X) = \underbrace{\hat{R}(\cdots(\hat{R}}_{n \text{ times}}(X)))$$

Intuitively, $\hat{R(X)}$ is the set of elements that can be inferred using proof trees of height at most n.
Now define set A by:

$$A = \hat{R}(\emptyset) \cup \hat{R}^2(\emptyset) \cup \cdots = \bigcup_{n \in \mathbb{N}} \hat{R}^n(\emptyset)$$

Intuitively, A is the set of elements that can be inferred using finite proof trees.

3.1.1 Properties of $\hat{R}$

It can be shown that:

1. $\hat{R}$ is monotonic. That is, $X \subseteq Y \Rightarrow \hat{R}(X) \subseteq \hat{R}(Y)$.
2. $\hat{R}(X \cup Y) \supseteq \hat{R}(X) \cup \hat{R}(Y)$. This follows from the monotonicity of $\hat{R}$ (left as exercise).
3. $\hat{R}(X \cap Y) \subseteq \hat{R}(X) \cap \hat{R}(Y)$. This follows from the monotonicity of $\hat{R}$ (left as exercise).

Lemma 3.1 $\forall n \geq 0.\ \hat{R}^n(\emptyset) \subseteq \hat{R}^{n+1}(\emptyset)$.

Proof.

We will proceed by induction on n.

Base case: Clearly $\emptyset \subseteq \hat{R}(\emptyset)$.

Induction step: We have $\hat{R}^n(\emptyset) \subseteq \hat{R}^{n+1}(\emptyset)$ by the Inductive Hypothesis. So by monotonicity of $\hat{R}$, we get $\hat{R}(\hat{R}^n(\emptyset)) \subseteq \hat{R}(\hat{R}^{n+1}(\emptyset))$. Hence $\hat{R}^{n+1}(\emptyset) \subseteq \hat{R}^{n+2}(\emptyset)$.

$\square$

3.1.2 Properties of A

It can be shown that:

1. A is closed. That is, $\hat{R}(A) \subseteq A$.
2. A is consistent. That is, $A \subseteq \hat{R}(A)$.
   (Note that 1 and 2 imply that A is a fixed point of $\hat{R}$. I.e. $\hat{R}(A) = A$.)
3. A is the *least* fixed point of $\hat{R}$.

Proof of 1

Let $x \in \hat{R}(A)$.

Then, by definition of $\hat{R}(A)$, there exist $x_1, \cdots, x_k \in A$ s.t. $\dfrac{x_1 \quad \cdots \quad x_k}{x}$.

So, by definition of A, there are $n_1, \cdots, n_k \in \mathbb{N}$ s.t. $x_i \in \hat{R}^{n_i}(\emptyset)$ (for $1 \leq i \leq k$).

Let $n = \max(n_1, \cdots, n_k)$. Then for each $i = 1, \cdots, k$, we have $n_i \leq n$. By Lemma 3.1, $\hat{R}^{n_i}(\emptyset) \subseteq \hat{R}^n(\emptyset)$. Hence, $x_i \in \hat{R}^n(\emptyset)$ for each $i = 1, \cdots, k$. Using the rule:

$$\frac{x_1 \quad \cdots \quad x_k}{x}$$

we conclude that $x \in \hat{R}^{n+1}(\emptyset) \subseteq A$.

$\square$