CS611 Lecture 06 **Proofs by Induction** 10 September 2003

Scribe: Ruijie Wang, Xin Qi Lecturer: Radu Rugina

In this lecture, we are going to talk about an important proof technique: induction. First, we will talk about induction from a general perspective. Then, we discuss several kinds of proofs by induction.

1 Introduction to Induction

1.1 Inductively Defined Sets

We use inductive proofs to prove properties for inductively defined sets. There are several examples of inductively defined sets:

- The set of all natural numbers $\mathbb{N}$
- Syntactic constructs, such as **Aexp**, **Bexp**, and **Com** for IMP.
- Set of all proof trees, also referred to as *derivations* (e.g. for large step semantics)
- Command executions (e.g. for small step semantics)

Generally speaking, there are two kinds of elements in an inductively defined set:

- Basic elements: elements that are by default in the set. For instance, 0 for $\mathbb{N}$, constants and variables for **Aexp**, axioms for proof trees, etc.
- Inductive elements: elements defined in terms of other elements in the set. For instance, non-zero numbers for $\mathbb{N}$, additions, subtractions, and multiplications for **Aexp**, inductive rules for proof trees.

1.2 Constructing a Proof

Induction is used to prove that a property holds for all of the elements of an inductively defined set. At a general level, such a proof consists two steps:

Basis/Base Case Proof that the property holds for the basic elements, which is called basis or base case.

Inductive Step Proof that the property holds for the inductive elements, provided that it holds for the component elements used to construct this element. This assumption is called Induction Hypothesis.

2 Types of Induction

Now we are going to discuss different types of induction, each of which can be applied to some specific examples of inductively defined sets stated above.

2.1 Standard Induction

Standard Induction is used to prove properties for natural numbers $\mathbb{N}$. Suppose we have a property $P(n)$, and we want to prove that

$$\forall n \,.\, P(n)$$

The proof consists of the following parts:

- Base case: Prove that $P(0)$ holds.
- Inductive step: Prove that $\forall n \geq 0 \,.\, P(n) \Rightarrow P(n+1)$.
  Here, $P(n)$ is the Induction Hypothesis, denoted I.H. for short.

2.2 Course-of-value Induction

This is another kind of induction used to prove properties of natural numbers $\mathbb{N}$. The base case is the same as in Standard Induction, but the Inductive Step is slightly different:

- Base case: Prove that $P(0)$ holds.
- Inductive step: $\forall n \geq 0 \; . \; [(\forall k \leq n \; . \; P(k)) \Rightarrow P(n+1)]$
  Here, the Induction Hypothesis is $\forall k \leq n \; . \; P(k)$.

Note that the base case and the inductive step can be combined into one formula:

$$\forall n \geq 0 \; . \; [(\forall n < n, P(k)) \Rightarrow P(n)]$$

Here, when $n = 0$, there is no k such that $k < n$, so the formula $\forall k < n \; . \; P(k)$ trivially evaluates to true. Therefore, to prove the above implication for $n = 0$, we must prove $P(0)$.

We have the following question: is the Course-of-value Induction stronger than the Standard Induction? That is, is there something the Course-of-value Induction can prove, but the Standard Induction cannot? The answer is no. To see that, suppose we have a property $P(n)$ that we want to prove. Then we can define the following property $P'(n)$:

$$P'(n) = \forall k \leq n \; . \; P(k)$$

To prove $P(n)$, we can use standard induction for $P'(n)$. That is equivalent to proving $P(n)$ using course-of-values induction.

2.3 Structural Induction

Structural Induction is a technique aimed at proving properties for syntactic sets, using a proof that follows the structure of the grammar which define those sets. For example, consider the set of arithmetic expressions **Aexp**, defined by the following grammar:

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

where n and X are basic elements, and the others are inductive elements.

Suppose we want to prove that:

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

A proof by structural induction for this fact consists of proving the following:

- Base cases:

$$\begin{aligned} &\forall n \in \mathbb{Z} \; . \; P(n) \\ &\forall X \in \mathbf{Loc} \; . \; P(X) \end{aligned}$$

- Inductive steps:

$$\begin{aligned} \forall a_0, a_1 \in \mathbf{Aexp} \; . \; P(a_0) \wedge P(a_1) \quad &\Rightarrow \quad P(a_0 + a_1) \\ \forall a_0, a_1 \in \mathbf{Aexp} \; . \; P(a_0) \wedge P(a_1) \quad &\Rightarrow \quad P(a_0 - a_1) \\ \forall a_0, a_1 \in \mathbf{Aexp} \; . \; P(a_0) \wedge P(a_1) \quad &\Rightarrow \quad P(a_0 * a_1) \end{aligned}$$

Note that there are five cases in the proof, corresponding to the five productions in the grammar. In other words, the proof follows the syntactic structure defined by the grammar (hence the name of "structural induction").

Example. Prove that the evaluation of $a \in \mathbf{Aexp}$ is deterministic. That is, prove that:

$$\forall a \; . \; \forall \sigma \; . \; (\forall n, n' \; . \; \langle a, \sigma \rangle \to n \wedge \langle a, \sigma \rangle \to n' \Rightarrow n = n')$$

Hence, the property $P(a)$ that we want to prove for all expressions $a \in \mathbf{Aexp}$ is:

$$P(a) \equiv \forall \sigma \;.\; (\forall n, n' \;.\; \langle a, \sigma \rangle \to n \wedge \langle a, \sigma \rangle \to n' \Rightarrow n = n')$$

Proof. The proof consists of the following cases:

- **Case**: $a \equiv m$. We want to prove that:

$$\forall \sigma \;.\; (\forall n, n' \;.\; \langle m, \sigma \rangle \to n \wedge \langle m, \sigma \rangle \to n' \Rightarrow n = n')$$

  Let σ, m, n, n' such that: $\langle m, \sigma \rangle \to n$ and $\langle m, \sigma \rangle \to n'$. We want to show that $n = n'$.

  Since $\langle m, \sigma \rangle \to n$, there should be a (finite) proof tree with this conclusion. By inspecting all the semantic rules, the only rule that matches this conclusion is the axiom for constants:

$$\frac{}{\langle n, \sigma \rangle \to n}$$

  Therefore, we must have $m = n$. Using a similar reasoning, the fact that $\langle m, \sigma \rangle \to n'$ implies that $m = n'$. We therefore conclude that $n = n'$.

- **Case**: $a \equiv X$. We want to prove that

$$\forall \sigma \;.\; (\forall n, n' \;.\; \langle X, \sigma \rangle \to n \wedge \langle X, \sigma \rangle \to n' \Rightarrow n = n')$$

  Let σ, m, n, n' such that: $\langle X, \sigma \rangle \to n$ and $\langle X, \sigma \rangle \to n'$. We want to show that $n = n'$.

  Since $\langle X, \sigma \rangle \to n$, there should be a (finite) proof tree with this conclusion. By inspecting all the semantic rules, the only rule that matches this conclusion is the axiom for variables:

$$\frac{}{\langle X, \sigma \rangle \to \sigma(X)}$$

  Therefore, we must have $n = \sigma(X)$. Using a similar reasoning, the fact that $\langle X, \sigma \rangle \to n'$ implies that $n' = \sigma(X)$. We therefore conclude that $n = n'$.

- **Case**: $a \equiv a_0 + a_1$. We want to prove that

$$\forall \sigma \;.\; (\forall n, n' \;.\; \langle a_0 + a_1, \sigma \rangle \to n \wedge \langle a_0 + a_1, \sigma \rangle \to n' \Rightarrow n = n')$$

  Let σ, m, n, n' such that: $\langle a_0 + a_1, \sigma \rangle \to n$ and $\langle a_0 + a_1, \sigma \rangle \to n'$. We want to show that $n = n'$.

  Since $\langle a_0 + a_1, \sigma \rangle \to n$, there should be a (finite) proof tree with this conclusion. By inspecting all the semantic rules, the only rule that matches this conclusion is the rule for sum expressions. Hence, the proof tree has the following form:

$$\frac{\dfrac{\vdots}{\langle a_0, \sigma \rangle \to m_0} \quad \dfrac{\vdots}{\langle a_1, \sigma \rangle \to m_1}}{\langle a_0 + a_1, \sigma \rangle \to n} \text{ where } n = m_0 + m_1$$

  Similarly, since $\langle a_0 + a_1, \sigma \rangle \to n'$, there must exist a proof tree of the following form:

$$\frac{\dfrac{\vdots}{\langle a_0, \sigma \rangle \to p_0} \quad \dfrac{\vdots}{\langle a_1, \sigma \rangle \to p_1}}{\langle a_0 + a_1, \sigma \rangle \to n'} \text{ where } n' = p_0 + p_1$$

  These proof trees show that $\langle a_0, \sigma \rangle \to m_0$ and $\langle a_0, \sigma \rangle \to p_0$. By the induction hypothesis $P(a_0)$, we get $m_0 = p_0$. The above trees also show that $\langle a_1, \sigma \rangle \to m_1$ and $\langle a_1, \sigma \rangle \to p_1$. By the induction hypothesis $P(a_1)$, we get $m_1 = p_1$. Therefore $m_0 + m_1 = p_0 + p_1$, so $n = n'$.

- **Case**: $a \equiv a_0 - a_1$. The proof is similar proof as that for $a \equiv a_0 + a_1$.
- **Case**: $a \equiv a_0 * a_1$. The proof is similar proof as that for $a \equiv a_0 + a_1$.

Non-Example. Let's try to prove in a similar way that the evaluation of commands $c \in \mathbf{Com}$ is deterministic.

What we want to prove is

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

So the property $P(c)$ that we want to prove is:

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

Proof attempt. We will *try* to prove this property by structural induction. As we know, the grammar for commands is:

$$c ::= \mathbf{skip} \;\mid\; X := a \;\mid\; c_0; c_1 \;\mid\; \mathbf{if}\, b\, \mathbf{then}\, c_0\, \mathbf{else}\, c_1 \;\mid\; \mathbf{while}\, b\, \mathbf{do}\, c_0$$

Hence, the proof by structural induction must follow this rules. We will omit most of the cases, and just consider the case of the while loop: $c \equiv \mathbf{while}\, b\, \mathbf{do}\, c_0$. Let us denote this command by w. By induction hypothesis, we assume $P(c_0)$ holds. We want to prove $P(w)$.

Let σ, σ', and σ'' be states such that: $\langle c, \sigma \rangle \to \sigma'$ and $\langle c, \sigma \rangle \to \sigma''$. We want to prove that $\sigma' = \sigma''$.

Since $\langle w, \sigma \rangle \to \sigma'$, there should be a (finite) proof tree with this conclusion. By inspecting all the semantic rules, there are two rules that match this conclusion, the rules for while loops. Hence, the derivation for this relation is of one of the following forms:

$$\frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \mathbf{false}}}{\langle w, \sigma \rangle \to \sigma} \qquad \text{or} \qquad \frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \mathbf{true}} \quad \dfrac{\vdots}{\langle c_0, \sigma \rangle \to \sigma_0} \quad \dfrac{\vdots}{\langle w, \sigma_0 \rangle \to \sigma'}}{\langle w, \sigma \rangle \to \sigma'}$$

Similarly, the derivation for $\langle w, \sigma \rangle \to \sigma''$ must be of one of the following forms:

$$\frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \mathbf{false}}}{\langle w, \sigma \rangle \to \sigma} \qquad \text{or} \qquad \frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \mathbf{true}} \quad \dfrac{\vdots}{\langle c_0, \sigma \rangle \to \sigma_1} \quad \dfrac{\vdots}{\langle w, \sigma_1 \rangle \to \sigma''}}{\langle w, \sigma \rangle \to \sigma''}$$

We see that in all of the possible cases, the condition b is being evaluated in state σ. This evaluation can yield to either *true* or *false*. We consider these two sub cases separately.

- Subcase: $\langle b, \sigma \rangle \to$ *false*. Then the proof trees for $\langle w, \sigma \rangle \to \sigma$ and $\langle w, \sigma \rangle \to \sigma'$ will be the ones on the left. Therefore $\sigma' = \sigma$ and $\sigma'' = \sigma$. As a result, $\sigma' = \sigma''$.
- Subcase: $\langle b, \sigma \rangle \to$ *true*. Then the proof trees for $\langle w, \sigma \rangle \to \sigma$ and $\langle w, \sigma \rangle \to \sigma'$ are the ones on the right. By inspecting the premises in these trees, we determine that $\langle c_0, \sigma \rangle \to \sigma_0$ and $\langle c_0, \sigma \rangle \to \sigma_1$. By the induction hypothesis $P(c_0)$, we get that $\sigma_0 = \sigma_1$.

  The above trees also show that $\langle w, \sigma_0 \rangle \to \sigma'$ and $\langle w, \sigma_1 \rangle \to \sigma''$. We already know that $\sigma_0 = \sigma_1$. We would like to conclude from here that $\sigma' = \sigma''$. But this is exactly what we are trying to prove in the first place! In other words, we reduced the problem to itself and haven't made any progress!

Therefore, the proof cannot be completed. Structural induction is not powerful enough to prove this property. The limitation here is that we have the same piece of syntax ($\mathbf{while}\, b\, \mathbf{do}\, c_0$) in the premise and in the conclusion of a rule. What we would like is an inductive proof based on the height of the derivations, because the derivation for w in the premise is smaller than the derivation for w in the conclusion. This kind of induction is called *induction on derivations*.

2.4 Induction on Derivations

This kind of induction is aimed at proving that a property $P(d)$ holds for each derivation (proof tree) d:

$$\forall d \,.\, P(d)$$

We need to introduce an ordering between derivations. We write $d \prec d'$, and say that d is less than d', is d is a sub-derivation (a subtree) of d'. Then, an inductive proof on derivations consists of proving the following:

- Base case: Prove that $P(d)$ holds, for all axioms d.
- Inductive case: Prove that $P(d)$ holds, provided that $P(d')$ holds for all sub-derivations $d' \prec d$.

To be continued in next lecture...