Perhaps the single most important proof technique in the theory of programming languages is *mathmatical induction*. In this lecture we will take a very general look at the subject and identify necessary and sufficient conditions under which the induction principle is valid. We will define *induction on a well-founded relation*, illustrate the definition with several examples, and then take another look at our inference rules in this light.

1 Free Variables Revisited

Recall that some of the substitution rules mentioned the set of free variables $FV(e)$ that occur in the expression e.

$$\begin{array}{lcll} (\lambda y.\, e_0)\{e_1/x\} & = & \lambda y.\, (e_0\{e_1/x\}), & \text{where } y \neq x \text{ and } y \notin FV(e_1), \\ (\lambda y.\, e_0)\{e_1/x\} & = & \lambda z.\, (e_0\{z/y\}\{e_1/x\}), & \text{where } z \neq x,\ z \notin FV(e_0), \text{ and } z \notin FV(e_1). \end{array}$$

Let us examine the definition of the free-variable function $FV : \{\lambda\text{-terms}\} \to \mathit{Var}$.

$$FV(x) = \{x\} \qquad FV(e_1\, e_2) = FV(e_1) \cup FV(e_2) \qquad FV(\lambda x.\, e) = FV(e) - \{x\}.$$

Why does this definition uniquely determine the function FV? There are two issues here:

- *existence*: whether FV is defined on all λ-terms;
- *uniqueness*: whether the definition is unique.

Of relevance is the fact that there are three clauses in the definition of FV corresponding to the three clauses in the definition of λ-terms and that a λ-term can be formed in one and only one way by one of these three clauses. Note also that although the symbol FV occurs on the right-hand side in two of these three clauses, they are applied to proper (*proper* = strictly smaller) subterms.

The idea underlying this definition is called *structural induction*. This is an instance of a general induction principle called *induction on a well-founded relation*.

2 Well-Founded Relations

A binary relation $\prec$ is said to be *well-founded* if it has no infinite descending chains. An *infinite descending chain* is an infinite sequence of elements $a_0, a_1, a_2, \ldots$ such that $a_{i+1} \prec a_i$ for all $i \geq 0$. Note that a well-founded relation cannot be reflexive.

Here are some examples of well-founded relations:

- the successor relation $\{(m, m+1) \mid m \in \mathbb{N}\}$ on $\mathbb{N}$;
- the less-than relation $<$ on $\mathbb{N}$;
- the element-of relation $\in$ on sets. The axiom of foundation (or axiom of regularity) of Zermelo–Fraenkel (ZF) set theory implies that $\in$ is well-founded. Among other things, this prevents a set from being a member of itself;
- the proper subset relation $\subset$ on the set of finite subsets of $\mathbb{N}$.

The following are not well-founded relations:

- the predecessor relation $\{(m+1, m) \mid m \in \mathbb{N}\}$ on $\mathbb{N}$ ($0, 1, 2, \ldots$ is an infinite *descending* chain!);
- the greater-than relation $>$ on $\mathbb{N}$;
- the less-than relation $<$ on $\mathbb{Z}$ ($0, -1, -2, \ldots$ is an infinite descending chain);
- the less-than relation $<$ on the real interval $[0, 1]$ ($1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$ is an infinite descending chain);
- the proper subset relation $\subset$ on subsets of $\mathbb{N}$ ($\mathbb{N}, \mathbb{N} - \{0\}, \mathbb{N} - \{0, 1\}, \ldots$ is an infinite descending chain).

3 Well-Founded Induction

Let $\prec$ be a well-founded binary relation on a set A. Abstractly, a *property* is just a map $P : A \to \mathbb{2}$, or equivalently, a subset $P \subseteq A$ (the set of all $a \in A$ for which $P(a) = \mathit{true}$).

The principle of well-founded induction on the relation $\prec$ says that in order to prove that a property P holds for all elements of A, it suffices to prove that P holds of any $a \in A$ whenever P holds for all $b \prec a$. In other words,

$$\forall a \in A\ (\forall b \in A\ b \prec a \Rightarrow P(b)) \Rightarrow P(a) \quad \Rightarrow \quad \forall a \in A\ P(a). \tag{1}$$

Expressed as a proof rule,

$$\frac{\forall a \in A\ (\forall b \in A\ b \prec a \Rightarrow P(b)) \Rightarrow P(a)}{\forall a \in A\ P(a)}. \tag{2}$$

The basis of the induction is when a has no $\prec$-predecessors; in that case, the premise $\forall b \in A\ b \prec a \Rightarrow P(b)$ is vacuously true.

For the well-founded relation $\{(m, m+1) \mid m \in \mathbb{N}\}$, (1) and (2) reduce to the familiar notion of mathematical induction on $\mathbb{N}$: to prove $\forall n\ P(n)$, it suffices to prove that $P(0)$ and that $P(n+1)$ whenever $P(n)$.

For the well-founded relation $<$ on $\mathbb{N}$, (1) and (2) reduce to *strong* induction on $\mathbb{N}$: to prove $\forall n\ P(n)$, it suffices to prove that $P(n)$ whenever $P(0), P(1), \ldots, P(n-1)$. When $n = 0$, the induction hypothesis is vacuously true.

3.1 Equivalence of Well-Foundedness and the Validity of Induction

In fact, one can show that the induction principle (1)–(2) is valid for a binary relation $\prec$ on A if and only if $\prec$ is well-founded.

To show that well-foundedness implies the validity of the induction principle, suppose for a contradiction that the induction principle were not valid. Then there would exist a property P for which the premise of (2) holds but not the conclusion. Thus $P(a_0)$ would false for some element $a_0 \in A$. The premise of (2) is equivalent to

$$\forall a \in A\ \neg P(a) \Rightarrow \exists b \in A\ b \prec a \wedge \neg P(b);$$

this implies that there exists $a_1 \prec a_0$ such that $P(a_1)$ is false. But since $P(a_1)$ is false, for the same reason there exists $a_2 \prec a_1$ such that $P(a_2)$ is false. Continuing in this fashion, using the axiom of dependent choice (a weak form of the axiom of choice), one can construct an infinite descending chain $a_0, a_1, a_2, \ldots$ for which $P(a_i)$ is false for all $i \geq 0$, so $\prec$ is not well-founded.

Conversely, suppose that there exists an infinite descending chain $a_0, a_1, a_2, \ldots$. Then the property $P(a)$ that says "$a \notin \{a_0, a_1, a_2, \ldots\}$" violates (2), since the premise of (2) holds but not the conclusion.

4 Structural Induction

Now let us define a well-founded relation on the set of all λ-terms. Define $e < e'$ if e is a *proper* subterm of e'. A λ-term e is a *proper* (or *strict*) subterm of e' if it is a subterm of e' and if $e \neq e'$. If we think of λ-terms as finite labeled trees, then e' is a tree that has e as a subtree. Since these trees are finite, the relation is well-founded. Induction on this relation is called *structural induction*.

We can now show that $FV(e)$ exists and is uniquely defined for any λ-term e. In the grammar for λ-terms, for any e, exactly one case in the definition of FV applies to e, and all references in the definition of FV are to subterms, which are strictly smaller. The function FV exists and is uniquely defined for the base case of the smallest λ-terms $x \in Var$. So $FV(e)$ exists and is uniquely defined for any λ-term e by induction on the well-founded subexpression relation.

We often have a set of expressions in a language built from a set of *constructors* starting from a set of *generators*. For example, in the case of λ-terms, the generators are the variables $x \in Var$ and the constructors are the binary application operator $\cdot$ and the unary abstraction operators λx. The set of expressions defined by the generators and constructors is the smallest set containing the generators and closed under the constructors.

If a function is defined on expressions in such a way that

- there is one clause in the definition for every generator or constructor pattern,
- the right-hand sides refer to the value of the function only on proper subexpressions,

then the function is well-defined and unique.

5 Inference Rules

We defined small-step semantics using inference rules. These rules are another kind of inductive definition. To prove properties of them, we would like to use well-founded induction.

To do this, we can change our view and look at reduction as a binary relation. To say that $e \to e'$ according to the small-step rules just means that the pair (e, e') is a member of some reduction relation, which is a subset of $\mathsf{Exp} \times \mathsf{Exp}$. In fact, not only is it a relation, it is a partial function.

For example, one of the small-step rules for call-by-value λ-calculus is

$$\frac{}{(\lambda x.\, e)\, v \to e\{v/x\}} \qquad \frac{e_1 \to e_1'}{e_1\, e_2 \to e_1'\, e_2} \qquad \frac{e \to e'}{v\, e \to v\, e'} \tag{3}$$

Here e_1, e_2, e_1' are *metavariables*. Expressions appearing above the line are called *premises*, and the expression below the line is called the *conclusion*. Sometimes one also sees an expression in parentheses to the right of the rule; this is called a *side condition* and represents a restriction on when the rule can be applied.[1]

A *rule instance* is a substitution for all the metavariables that satisfies the side condition. For example, here is an instance of the rule (3):

$$\frac{(\lambda x.\, x)(\lambda z.\, z) \to (\lambda z.\, z)}{((\lambda x.\, x)(\lambda z.\, z))\Omega \to (\lambda z.\, z)\Omega}$$

[1] Side conditions and premises should not be confused. The difference is that side conditions are not part of the relation that the rule is trying to define, whereas the premises are.

With rules like (3), we are usually trying to define some set or relation by induction. For example, the rule (3) is part of the inductive definition of the reduction relation $\rightarrow$, which is a subset of $\mathsf{Exp} \times \mathsf{Exp}$. Such rules are typically of the form

$$\frac{X_1 \; X_2 \; \ldots \; X_n}{X} \, (\varphi), \tag{4}$$

where $X_1, \ldots, X_n$ specify elements that are already members of the set or relation being defined and X represents a new element constructed from $X_1, \ldots, X_n$ that is to be added to the relation. The side condition φ, which may or may not be present, is a condition on $X_1, \ldots, X_n$ and X that must hold for the rule to be applicable.

Now suppose we have written down a set of rules in an attempt to define a set or relation A. How do we know whether A is well-defined? If the rule (4) is in force, then surely we would like to have $X \in A$ whenever $X_1, \ldots, X_n \in A$ and the side condition, if any, holds; but there may be many sets A for which this is true, so this is hardly a definition.

6 Set Operators

To see how the definition works, we can view inference rules as *monotone set operators*. Suppose we have a rule R of the form (4). Thus the X and the X_j in (4) represent members of some set S. We can view R as a mapping on subsets of S. Given $B \subseteq S$, define

$$R(B) \;\triangleq\; \{X \mid \{X_1, X_2, \ldots, X_n\} \subseteq B \text{ and } \frac{X_1 \; X_2 \; \ldots \; X_n}{X} \text{ is an instance of (4)}\}.$$

Then R is a function mapping subsets of S to subsets of S; that is, $R : 2^S \rightarrow 2^S$, where 2^S denotes the *powerset* (set of all subsets) of S.

Now suppose we have a finite set of such rules $R_1, \ldots, R_m$. What set $A \subseteq S$ is defined by the rules? At the very least, we would like A to satisfy the following two properties:

- A is *R-closed*: $R_1(A) \cup \cdots \cup R_m(A) \subseteq A$. We would like this to hold because we would like every element that the rules say should be included in A are actually included in A.
- A is *R-consistent*: $A \subseteq R_1(A) \cup \cdots \cup R_m(A)$. We would like this to hold because we would like every element of A to be included in A *only* as a result of applying one of the rules.

These two properties together say that $A = R_1(A) \cup \cdots \cup R_n(A)$, or in other words, A should be a *fixed point* of the set map $\lambda A.\, R_1(A) \cup \cdots \cup R_n(A)$. This leads to two natural questions:

- Does this map actually have a fixed point?
- Is the fixed point unique? If not, which one should we take?

We will answer these questions next time.