CS 611: Lecture 6

Rule Induction

September 8, 1999

Cornell University Computer Science Department
Andrew Myers

Administration

- No class on Friday
- Homework #1 due on Monday in class (implementation due at 5 PM Monday)

Induction

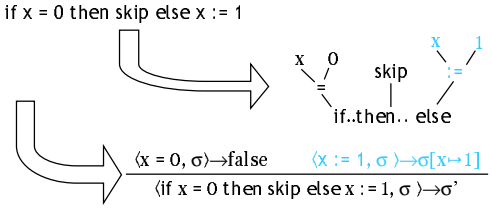
- Last time: two new induction techniques for proving properties of programs
- Structural induction:
  - prove that a property holds of all language atoms
  - prove that it holds for each kind of expression if it holds of the parts of the expression

  ⇒ property holds for *all* expressions
- Induction on derivations
  - prove it holds for derivations that are axioms
  - prove property holds if it holds for every *derivation* (evaluation) of parts of an expression

  ⇒ property holds for *all* derivations

Observation

- These two forms of induction are very similar — both operate on trees.



Expression inference rules

BNF spec for arithmetic expressions in IMP:

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

Let A be the set of all arithmetic expressions.
Inductive definition of A via inference rules:

Axioms: $\dfrac{}{n \in A}$ $\dfrac{}{X \in A}$

Rules: $\dfrac{a_0 \in A \quad a_1 \in A}{a_0 + a_1 \in A}$ $\dfrac{a_0 \in A \quad a_1 \in A}{a_0 - a_1 \in A}$

$$\frac{a_0 \in A \quad a_1 \in A}{a_0 \times a_1 \in A}$$

Expression derivation tree

- Every legal expression now has a derivation tree.
- Example: (2+3) × (4-x)

$$\frac{\dfrac{2 \in A \quad 3 \in A}{2+3 \in A} \quad \dfrac{4 \in A \quad x \in A}{4 - 5 \in A}}{(2+3) \times (4 - 5) \in A}$$

Proof systems = set definitions

- Rules defining the set of all legal arithmetic expressions:

$$\frac{}{n \in A} \qquad \frac{}{X \in A}$$

$$\frac{a_0 \in A \quad a_1 \in A}{a_0 \times a_1 \in A} \qquad \frac{a_0 \in A \quad a_1 \in A}{a_0 + a_1 \in A} \qquad \frac{a_0 \in A \quad a_1 \in A}{a_0 - a_1 \in A}$$

- Can view as constraint equations on A:

$$A \supseteq \mathrm{Z} \qquad A \supseteq \mathrm{Loc}$$
$$A \supseteq \{ a_0 \times a_1 \mid a_0 \in A \wedge a_1 \in A \}$$
$$A \supseteq \{ a_0 + a_1 \mid a_0 \in A \wedge a_1 \in A \}$$
$$A \supseteq \{ a_0 - a_1 \mid a_0 \in A \wedge a_1 \in A \}$$

Solving recursive equations

$$A \supseteq \mathrm{Z} \qquad A \supseteq \mathrm{Loc}$$
$$A \supseteq \{ a_0 \times a_1 \mid a_0 \in A \wedge a_1 \in A \}$$
$$A \supseteq \{ a_0 + a_1 \mid a_0 \in A \wedge a_1 \in A \}$$
$$A \supseteq \{ a_0 - a_1 \mid a_0 \in A \wedge a_1 \in A \}$$

- A is *not the only set* that satisfies these equations!
- Can we make these equations mean what we want by a rule for picking among the possible sets?

Rule application operator

- Define operator $\mathbf{R}(A')$ that adds all elements needed to satisfy constraint equations:

$$A \supseteq \mathrm{Z} \qquad A \supseteq \mathrm{Loc}$$
$$A \supseteq \{ a_0 + a_1 \mid a_0 \in A \wedge a_1 \in A \}$$
$$A \supseteq \{ a_0 - a_1 \mid a_0 \in A \wedge a_1 \in A \}$$
$$A \supseteq \{ a_0 \times a_1 \mid a_0 \in A \wedge a_1 \in A \}$$

$$\mathbf{R}(A') = A' \cup \mathrm{Z} \cup \mathrm{Loc} \cup \{ a_0 + a_1 \mid a_0 \in A' \wedge a_1 \in A' \} \cup \{ a_0 - a_1 \mid a_0 \in A' \wedge a_1 \in A' \} \cup \{ a_0 \times a_1 \mid a_0 \in A' \wedge a_1 \in A' \}$$

$$A = \mathbf{R}(A)$$

Want the *least fixed point* of $\mathbf{R}$

Constructing least fixed point

- Assuming we have A' containing only elements of A (A')
- $\mathbf{R}(A')$ contains only elements of A too
- $\mathbf{R}(\mathbf{R}(A'))$ contains only elements of A too
- Empty set $\varnothing$ contains only elements of A, so

$$A \supseteq \ldots \supseteq \mathbf{R}^{n+1}(\varnothing) \supseteq \mathbf{R}^{n}(\varnothing) \supseteq \mathbf{R}^{n-1}(\varnothing) \supseteq \ldots \supseteq \mathbf{R}(\mathbf{R}(\varnothing)) \supseteq \mathbf{R}(\varnothing) \supseteq \varnothing$$

- Therefore, $A = \bigcup_{n \in \omega} \mathbf{R}^n(\varnothing)$

$A = fix\,(\mathbf{R})$ — *Least fixed point* operator

Proof by rule induction

$$A = fix(\mathbf{R}) = \bigcup_{n \in \omega} \mathbf{R}^n(\varnothing)$$

- For every element E of A, there is some smallest i such that $E \in \mathbf{R}^i(\varnothing)$
- i is the minimum number of times the inference rules must be applied to construct E : height of derivation
- Goal: prove that property P holds for every expression in A : $\forall a \in A \,.\, \mathrm{P}(A)$
- Base case: prove $\forall\, a \in \mathbf{R}(\varnothing) \,.\, \mathrm{P}(a)$
- Inductive step: prove $\forall n>1, a \in \mathbf{R}^n(\varnothing) \,.\, (\forall a' \in \mathbf{R}^{n-1}(\varnothing) \,.\, \mathrm{P}(a')) \Rightarrow \mathrm{P}(a)$
- Conclusion: $\forall a \in fix\,(\mathbf{R}) \,.\, \mathrm{P}(a)$

Proof recipe

- $\mathbf{R}$ is defined by some set of rules
- Rules look like $\frac{}{x}$ *or* $\frac{x_1,\ldots,x_n}{x}$
- Want to show that $\forall a \in \mathbf{R}(\varnothing) \,.\, \mathrm{P}(a)$
  $\Rightarrow$technique: show $\mathrm{P}(x)$ for all axioms $\frac{}{x}$
- Want to show $\forall n>1, a \in \mathbf{R}^n(\varnothing) \,.\, (\forall a' \in \mathbf{R}^{n-1}(\varnothing) \,.\, \mathrm{P}(a')) \Rightarrow \mathrm{P}(a)$
  $\Rightarrow$technique: for all rules $\frac{x_1,\ldots,x_n}{x}$ show

$$(\forall x' \in \{x_1,\ldots,x_n\} \,.\, \mathrm{P}(x')) \Rightarrow \mathrm{P}(x)$$

Another rule induction instance

- Structural induction = instance of general *rule induction* technique, based on *inductive definition* of set of legal expressions
- What about operational semantics? Is a set being defined inductively?

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

- Let I_R be the set of all legal evaluations of arithmetic expressions

$\langle a, \sigma\rangle \to n$ **means** $(a, \sigma, n) \in I_R$

Completing the mapping

- Evaluation inference rules are an inductive definition of the set of legal evaluations I_R

$$\frac{\langle a_0, \sigma\rangle \to n_0 \qquad \langle a_1, \sigma\rangle \to n_1}{\langle a_0 + a_1, \sigma\rangle \to n} \quad \text{(where } n = n_0 + n_1\text{)}$$

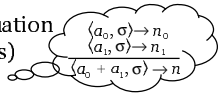
$$\Updownarrow$$

$$\frac{(a_0, \sigma, n_0) \in I_R \qquad (a_1, \sigma, n_1) \in A}{(a_0 + a_1, \sigma, n) \in I_R} \quad \text{(where } n = n_0 + n_1\text{)}$$

$\mathbf{R}(B) = B \cup \ldots \cup$
$\{ (a_0 + a_1, \sigma, n) \mid \exists n_0, n_1 .$
$(a_0, \sigma, n_0) \in B \wedge (a_1, \sigma, n_1) \in B \wedge n = n_0 + n_1 \} \cup \ldots$

Proof technique

- To prove property P holds for all evaluations $\langle a, \sigma\rangle \to n$, show
  - It holds for axioms (evaluation of constants and locations)
  - For each evaluation rule,
    - Assume P holds for all antecedent evaluations
    - Under this assumption, prove it holds for the conclusion of the rule



- Rule induction: property P holds for all evaluations $\langle a, \sigma\rangle \to n$

Induction on derivations

$$\frac{x^j_1, \ldots, x^j_{nj}}{x^j}$$

- Let $\{R_j\}$ be a set of rules defining some set A
- A derivation d using these rules is a legal derivation if it can be constructed from some rule R_j and sub-derivations d^j_i for each of the antecedents x^j_i
- Thus, we can define inference rules for constructing set I_R of *derivations* of elements of A
- Apply rule induction proof recipe to find proofs that properties P hold for all legal derivations : induction on derivations
  - P(d) holds for all one-step derivations d
  - P(d) holds for all n-step derivations assuming it holds for all shorter derivations

Summary

- Any proof system (set of inference rules) is an inductive definition of a set
- Rule induction can be applied to any inductive definition
- Examples: structural induction, induction on derivations are both instances of this approach
- We will use rule induction for other proof systems in course (*e.g.*, type-checking rules)