CIS 500

Software Foundations

Fall 2005

September 14

Announcements

I will be away September 19-October 5.

- I will be reachable by email.
- Fastest response—cis500@cis.upenn.edu
- No office hours 9/19, 9/26, 10/3
- Guest lecturers for the next 3 weeks.

Well-founded induction

Induction principles

We've seen three definitions of sets and their associated induction principles:

- Natural numbers
- Boolean terms
- Arithmetic terms

Given a set defined with BNF, it is not too hard to describe the structural induction principle for that set.

For example:

```
t ::= brillig
      tove
      snicker t
      gyre t gimble t
```

What is the structural induction principle for this language?

A Question

Why are any of these induction principles true? Why should I believe a proof that employs one?

Well-founded induction

Well-founded induction is a generalized form of all of these induction principles.

Let $\prec$ be a well-founded relation on a set A. Let P be a property. Then $\forall a \in A.P(a)$ iff

$$\forall a \in A.([\forall b \prec a.P(b)] \Rightarrow P(a))$$

Choosing the right set A and relation $\prec$ determines the induction principle.

Well-founded induction

For example, we let $A = \mathcal{N}$ and $n \prec m \stackrel{\text{def}}{=} m = n + 1$. In this case, we can rewrite previous principle as:

$\forall a \in \mathcal{N}.P(a)$ iff

$$\forall a \in \mathcal{N}.([\forall b \prec a.P(b)] \Rightarrow P(a))$$

Now, by definition a is either 0 or $i + 1$ for some i:

$\forall a \in \mathcal{N}.P(a)$ iff

$$[\forall b \prec 0.P(b)] \Rightarrow P(0) \land$$
$$\forall i \in \mathcal{N}.[\forall b \prec i + 1.P(b)] \Rightarrow P(i + 1)$$

Simplify to:

$\forall a \in \mathcal{N}.P(a)$ iff $P(0) \land \forall i \in \mathcal{N}.P(i) \Rightarrow P(i + 1)$

Strong induction

If $\prec$ is the "strictly less than" relation $<$, then the principle we get is strong induction.

$\forall a \in \mathcal{N}.P(a)$ iff

$$\forall a \in \mathcal{N}.([\forall b < a.P(b)] \Rightarrow P(a))$$

Well-founded relation

The induction principle holds only when the relation $\prec$ is well-founded.

Definition: A well-founded relation is a binary relation $\prec$ on a set A such that there are no infinite descending chains $\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$.

Are the successor and $<$ relations well-founded?

Structural induction

Well-founded induction also generalizes structural induction.

If $\prec$ is the "immediate subterm" relation for an inductively defined set, then the principle we get is structural induction.

For example, in Arith, the term t_1 is an immediate subterm of the term `succ t1`.

Is the immediate subterm relation well-founded?

Proof of well-founded induction

We'd like to show that:

Theorem: Let $\prec$ is a well-founded relation on a set A. Let P be a property. Then $\forall a \in A.P(a)$ iff

$$\forall a \in A.([\forall b \prec a.P(b)] \Rightarrow P(a)$$

The ($\Rightarrow$) direction is trivial. We'll show the ($\Leftarrow$) direction.

First, observe that any nonempty subset Q of A has a minimal element, even if Q is infinite.

Now, suppose $\neg P(a)$ for some a in A. There must be a minimal element m of the set $\{a \in A | \neg P(a)\}$. But then, $\neg P(m)$ yet $[\forall b \prec m.P(b)]$ which is a contradiction.

Properties of small-step semantics

Small-step semantics

Booleans:

$$\texttt{if true then } t_2 \texttt{ else } t_3 \longrightarrow t_2 \qquad \texttt{if false then } t_2 \texttt{ else } t_3 \longrightarrow t_3$$

$$\frac{t_1 \longrightarrow t_1'}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \longrightarrow \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3}$$

Natural numbers:

$$\frac{t_1 \longrightarrow t_1'}{\texttt{succ } t_1 \longrightarrow \texttt{succ } t_1'} \quad \texttt{pred 0} \longrightarrow \texttt{0} \quad \texttt{pred (succ } nv_1\texttt{)} \longrightarrow nv_1 \quad \frac{t_1 \longrightarrow t_1'}{\texttt{pred } t_1 \longrightarrow \texttt{pred } t_1'}$$

Both:

$$\texttt{iszero 0} \longrightarrow \texttt{true} \quad \texttt{iszero (succ } nv_1\texttt{)} \longrightarrow \texttt{false} \quad \frac{t_1 \longrightarrow t_1'}{\texttt{iszero } t_1 \longrightarrow \texttt{iszero } t_1'}$$

Digression

Suppose we wanted to change our evaluation strategy so that the `then` and `else` branches of an `if` get evaluated (in that order) before the guard. How would we need to change the rules?

Digression

Suppose we wanted to change our evaluation strategy so that the `then` and `else` branches of an `if` get evaluated (in that order) before the guard. How would we need to change the rules?

Suppose, moreover that if the evaluation of the `then` and `else` branches leads to the same value, we want to immediately produce that value ("short-circuiting" the evaluation of the guard). How would we need to change the rules?

Digression

Suppose we wanted to change our evaluation strategy so that the `then` and `else` branches of an `if` get evaluated (in that order) before the guard. How would we need to change the rules?

Suppose, moreover that if the evaluation of the `then` and `else` branches leads to the same value, we want to immediately produce that value ("short-circuiting" the evaluation of the guard). How would we need to change the rules?

Of the rules we just invented, which are computation rules and which are congruence rules?

Normal forms

- A normal form is a term that cannot be evaluated any further – i.e. a term `t` is a normal form (or "is in normal form") is there is no `t′` such that $t \to t'$
- A normal form is a state where the abstract machine is halted – it can be regarded as a "result" of evaluation.
- The meaning of a term `t` with small-step semantics is a term `t′`, such that $t \to^* t'$ and `t′` is a normal form.

  We say that `t′` "is the normal form of" `t`.

Normal forms

- For Arith, not all normal forms are values, but every value is a normal form.
- A term like `succ false` that is a normal form, but is not a value, is "stuck".

Properties of this semantics

- (Homework): This small-step semantics "agrees" with the large-step semantics for terms that do not get stuck. In other words, $t \Downarrow v$ if and only if $t \to^* v$.
- The $\to$ relation is deterministic. If $t \to t'$ and $t \to t''$ then $t' = t''$.
- Evaluation is deterministic: There is at most one normal form for a term `t`. (Easy to prove: Follows because the $\to$ relation is deterministic).
- Evaluation is total: There is at least one normal form for a term `t`. (More difficult to prove: Must show that there are no infinite sequences of small-step evaluation.)

Reasoning about evaluation

Induction on evaluation

We can define an induction principle for small-step evaluation. Recall the definition (just for booleans, for now):

$$\texttt{if true then } t_2 \texttt{ else } t_3 \longrightarrow t_2 \qquad \text{(E-IfTrue)}$$

$$\texttt{if false then } t_2 \texttt{ else } t_3 \longrightarrow t_3 \qquad \text{(E-IfFalse)}$$

$$\frac{t_1 \longrightarrow t_1'}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \longrightarrow \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3} \qquad \text{(E-If)}$$

What is the induction principle for this relation?

Using this induction principle

For all t, t', $P(t \longrightarrow t')$ if

- $P(\texttt{if true then } t_2 \texttt{ else } t_3 \longrightarrow t_2)$ and
- $P(\texttt{if false then } t_2 \texttt{ else } t_3 \longrightarrow t_3)$ and
- $P(\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \longrightarrow \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3)$ given that $P(t_1 \longrightarrow t_1')$

What does it mean to say $P(\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \longrightarrow \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3)$?

Derivations

Another way to look at it is in terms of derivations.

A derivation records the "justification" for a particular pair of terms that are in the evaluation relation, in the form of a tree. We've all ready seen one example:

(example on the board)

Terminology:

- These trees are called derivation trees (or just derivations)
- The final statement in a derivation is the conclusion
- We say that a derivation is a witness for its conclusion (or a proof of its conclusion) – it records the reasoning steps to justify the conclusion
- When we reason about the conclusions, we are reasoning about derivations

Observation

Lemma: Suppose we are given a derivation $\mathcal{D}$ witnessing the pair (t, t') in the $\longrightarrow$ relation. Then either:

1. the final rule used in $\mathcal{D}$ is E-IfTrue and we have $t = \texttt{if true then } t_2 \texttt{ else } t_3$ and $t' = t_2$ for some t_2 and t_3, or
2. the final rule used in $\mathcal{D}$ is E-IfFalse and we have $t = \texttt{if false then } t_2 \texttt{ else } t_3$ and $t' = t_3$ for some t_2 and t_3, or
3. the final rule used in $\mathcal{D}$ is E-If and we have $t = \texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3$ and $t' = \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3$, for some t_1, t_1', t_2 and t_3; moreover the immediate subderivation of $\mathcal{D}$ witnesses $t_1 \longrightarrow t_1'$.

Induction on Derivations

We can now write proofs about evaluation "by induction on derivation trees."

Given an arbitrary derivation $\mathcal{D}$ with conclusion $\mathtt{t} \longrightarrow \mathtt{t}'$, we assume the desired result for its immediate sub-derivation (if any) and proceed by a case analysis (using the previous lemma) of the final evaluation rule used in constructing the derivation tree.

E.g....

Induction on small-step evaluation

For example, we can show that small-step evaluation is deterministic.

Theorem: If $\mathtt{t} \longrightarrow \mathtt{t}'$ then if $\mathtt{t} \longrightarrow \mathtt{t}''$ then $\mathtt{t}' = \mathtt{t}''$.

Proof: By induction on a derivation $\mathcal{D}$ of $\mathtt{t} \longrightarrow \mathtt{t}'$.

1. Suppose the final rule used in $\mathcal{D}$ is E-IfTrue, with $\mathtt{t} = \mathtt{if}\ \mathtt{t}_1\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3$ and $\mathtt{t}_1 = \mathtt{true}$ and $\mathtt{t}' = \mathtt{t}_2$. Therefore, the last rule of the derivation of $\mathtt{t} \longrightarrow \mathtt{t}'$ cannot be E-IfFalse, because $\mathtt{t}_1$ is not `false`. Furthermore, the last rule cannot be E-If either, because this rule requires that $\mathtt{t}_1 \longrightarrow \mathtt{t}_1'$, and `true` does not step to anything. So the last rule can only be E-IfTrue.
2. Suppose the final rule used in $\mathcal{D}$ is E-IfFalse, with $\mathtt{t} = \mathtt{if}\ \mathtt{false}\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3$ and $\mathtt{t}' = \mathtt{t}_3$. This case is similar to the previous.
3. Suppose the final rule used in $\mathcal{D}$ is E-If, with $\mathtt{t} = \mathtt{if} \mathtt{t}_1\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3$ and $\mathtt{t}' = \mathtt{if} \mathtt{t}_1'\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3$, where $\mathtt{t}_1 \longrightarrow \mathtt{t}_1'$ is witnessed by a derivation

$\mathcal{D}_1$. The last rule in the derivation of $\mathtt{t} \longrightarrow \mathtt{t}''$ can only be E-If, so it must be that $\mathtt{t}_1 \longrightarrow \mathtt{t}_1''$. By induction $\mathtt{t}_1' = \mathtt{t}_1''$ so $\mathtt{t}' = \mathtt{t}''$.

What principle to use?

We've proven the same theorem using two different induction principles.

Q: Which one is the best one to use?

A: The one that works.

For these simple languages, anything you can prove by induction on $\mathtt{t} \longrightarrow \mathtt{t}'$, you can prove by structural induction on $\mathtt{t}$. But that will not be the case for every language.

Termination of evaluation

Termination of evaluation

Theorem: For every t there is some normal form t′ such that $t \longrightarrow^* t'$.

An Inductive Definition

We can define the size of a term with the following relation:

$$
\begin{aligned}
size(\texttt{true}) &= 1\\
size(\texttt{false}) &= 1\\
size(\texttt{0}) &= 1\\
size(\texttt{succ } t_1) &= size(t_1) + 1\\
size(\texttt{pred } t_1) &= size(t_1) + 1\\
size(\texttt{iszero } t_1) &= size(t_1) + 1\\
size(\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3) &= size(t_1) + size(t_2) + size(t_3) + 1
\end{aligned}
$$

Note: this is yet more shorthand. How would we write this definition with inference rules?

Induction on Derivations — Another Example

Theorem: If $t \longrightarrow t'$ — i.e., if $(t, t') \in \longrightarrow$ — then $size(t) > size(t')$.

Proof: By induction on a derivation $\mathcal{D}$ of $t \longrightarrow t'$.

1. Suppose the final rule used in $\mathcal{D}$ is E-IfTrue, with $t = \texttt{if true then } t_2 \texttt{ else } t_3$ and $t' = t_2$. Then the result is immediate from the definition of *size*.
2. Suppose the final rule used in $\mathcal{D}$ is E-IfFalse, with $t = \texttt{if false then } t_2 \texttt{ else } t_3$ and $t' = t_3$. Then the result is again immediate from the definition of *size*.
3. Suppose the final rule used in $\mathcal{D}$ is E-If, with $t = \texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3$ and $t' = \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3$, where $(t_1, t_1') \in \longrightarrow$ is witnessed by a derivation $\mathcal{D}_1$. By the induction hypothesis, $size(t_1) > size(t_1')$. But then, by the definition of *size*, we have $size(t) > size(t')$.

Termination of evaluation

Theorem: For every t there is some normal form t' such that $t \longrightarrow^* t'$.

Proof:

Termination of evaluation

Theorem: For every t there is some normal form t' such that $t \longrightarrow^* t'$.

Proof:

- First, recall that single-step evaluation strictly reduces the size of the term:

  if $t \longrightarrow t'$, then $size(t) > size(t')$

- Now, assume (for a contradiction) that

  $t_0, t_1, t_2, t_3, t_4, \ldots$

  is an infinite-length sequence such that

  $t_0, \longrightarrow t_1, \longrightarrow t_2, \longrightarrow t_3, \longrightarrow t_4 \longrightarrow \cdots,$

- Then

  $size(t_0), size(t_1), size(t_2), size(t_3), size(t_4), \ldots$

  is an infinite, strictly decreasing, sequence of natural numbers.

- But such a sequence cannot exist — contradiction!

Termination Proofs

Most termination proofs have the same basic form:

Theorem: The relation $R \subseteq X \times X$ is terminating — i.e., there are no infinite sequences x_0, x_1, x_2, etc. such that $(x_i, x_{i+1}) \in R$ for each i.

Proof:

1. Choose
   - a well-founded set $(W, <)$ — i.e., a set W with a partial order $<$ such that there are no infinite descending chains $w_0 > w_1 > w_2 > \ldots$ in W
   - a function f from X to W
2. Show $f(x) > f(y)$ for all $(x, y) \in R$
3. Conclude that there are no infinite sequences x_0, x_1, x_2, etc. such that $(x_i, x_{i+1}) \in R$ for each i), since, if there were, we could construct an infinite descending chain in W.