

CIS 500  
Software Foundations  
Fall 2003

17 September

Administrivia

- ◆ Reading for (before!) next week's lectures: TAPL Chapter 5

Review (and a few more details)

Simple Arithmetic Expressions

The set  $\mathcal{T}$  of terms is defined by the following abstract grammar:

$t ::=$	terms
true	constant true
false	constant false
if t then t else t	conditional
0	constant zero
succ t	successor
pred t	predecessor
iszero t	zero test

## Inference Rule Notation

More explicitly: The set  $\mathcal{T}$  is the **smallest** set **closed** under the following rules.

$$\frac{\text{true} \in \mathcal{T}}{t_1 \in \mathcal{T}} \quad \frac{\text{false} \in \mathcal{T}}{t_1 \in \mathcal{T}} \quad \frac{0 \in \mathcal{T}}{t_1 \in \mathcal{T}}$$

$$\frac{t_1 \in \mathcal{T}}{\text{succ } t_1 \in \mathcal{T}} \quad \frac{t_1 \in \mathcal{T}}{\text{pred } t_1 \in \mathcal{T}} \quad \frac{t_1 \in \mathcal{T}}{\text{iszero } t_1 \in \mathcal{T}}$$

$$\frac{t_1 \in \mathcal{T} \quad t_2 \in \mathcal{T} \quad t_3 \in \mathcal{T}}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \in \mathcal{T}}$$

## Generating Functions

Each of these rules can be thought of as a **generating function** that, given some elements from  $\mathcal{T}$ , generates some other element of  $\mathcal{T}$ . Saying that  $\mathcal{T}$  is closed under these rules means that  $\mathcal{T}$  cannot be made any bigger using these generating functions — it already contains everything “justified by its members.”

$$\frac{\text{true} \in \mathcal{T}}{t_1 \in \mathcal{T}} \quad \frac{\text{false} \in \mathcal{T}}{t_1 \in \mathcal{T}} \quad \frac{0 \in \mathcal{T}}{t_1 \in \mathcal{T}}$$

$$\frac{t_1 \in \mathcal{T}}{\text{succ } t_1 \in \mathcal{T}} \quad \frac{t_1 \in \mathcal{T}}{\text{pred } t_1 \in \mathcal{T}} \quad \frac{t_1 \in \mathcal{T}}{\text{iszero } t_1 \in \mathcal{T}}$$

$$\frac{t_1 \in \mathcal{T} \quad t_2 \in \mathcal{T} \quad t_3 \in \mathcal{T}}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \in \mathcal{T}}$$

Let's write these generating functions explicitly.

$$F_1(\mathbf{U}) = \{\text{true}\}$$

$$F_2(\mathbf{U}) = \{\text{false}\}$$

$$F_3(\mathbf{U}) = \{0\}$$

$$F_4(\mathbf{U}) = \{\text{succ } t_1 \mid t_1 \in \mathbf{U}\}$$

$$F_5(\mathbf{U}) = \{\text{pred } t_1 \mid t_1 \in \mathbf{U}\}$$

$$F_6(\mathbf{U}) = \{\text{iszero } t_1 \mid t_1 \in \mathbf{U}\}$$

$$F_7(\mathbf{U}) = \{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \mid t_1, t_2, t_3 \in \mathbf{U}\}$$

Each one takes a set of terms  $\mathbf{U}$  as input and produces a set of “terms justified by  $\mathbf{U}$ ” as output.

If we now define a generating function for the whole set of inference rules (by combining the generating functions for the individual rules),

$$F(\mathbf{U}) = F_1(\mathbf{U}) \cup F_2(\mathbf{U}) \cup F_3(\mathbf{U}) \cup F_4(\mathbf{U}) \cup F_5(\mathbf{U}) \cup F_6(\mathbf{U}) \cup F_7(\mathbf{U})$$

then we can restate the previous definition of the set of terms  $\mathcal{T}$  like this:

**Definition:**

- ◆ A set  $\mathbf{U}$  is said to be “closed under  $F$ ” (or “ $F$ -closed”) if  $F(\mathbf{U}) \subseteq \mathbf{U}$ .
- ◆ The set of terms  $\mathcal{T}$  is the smallest  $F$ -closed set.  
(i.e., if  $\mathbf{O}$  is another set such that  $F(\mathbf{O}) \subseteq \mathbf{O}$ , then  $\mathcal{T} \subseteq \mathbf{O}$ .)

## Another definition by generating functions

Our alternate definition of the set of terms can also be stated using the generating function  $F$ :

$$\mathcal{S}_0 = \emptyset$$

$$\mathcal{S}_{i+1} = F(\mathcal{S}_i)$$

$$\mathcal{S} = \bigcup_i \mathcal{S}_i$$

Compare this definition of  $\mathcal{S}$  with the one we saw last time:

$$\mathcal{S}_0 = \emptyset$$

$$\begin{aligned} \mathcal{S}_{i+1} = & \{ \text{true, false, 0} \} \\ & \cup \{ \text{succ } t_1, \text{pred } t_1, \text{iszero } t_1 \mid t_1 \in \mathcal{S}_i \} \\ & \cup \{ \text{if } t_1 \text{ then } t_2 \text{ else } t_3 \mid t_1, t_2, t_3 \in \mathcal{S}_i \} \end{aligned}$$

$$\mathcal{S} = \bigcup_i \mathcal{S}_i$$

The only difference is that we have “pulled out”  $F$  and given it a name.

Note that our two definitions of terms characterize the same set from different directions:

- ◆ “from above,” as the intersection of all  $F$ -closed sets;
- ◆ “from below,” as the limit (union) of a series of sets that start from  $\emptyset$  and get “closer and closer to being  $F$ -closed.”

Proposition 3.2.6 in the book shows that these two definitions actually define the same set.

**Warning:** Hard hats on for the next slide!

## Structural Induction

The principle of structural induction on terms can also be re-stated using generating functions:

Suppose  $T$  is the smallest  $F$ -closed set.

If, for each set  $U$ ,

from the assumption “ $P(u)$  holds for every  $u \in U$ ”

we can show “ $P(v)$  holds for any  $v \in F(U)$ ,”

then  $P(t)$  holds for all  $t \in T$ .

## Structural Induction

The principle of structural induction on terms can also be re-stated using generating functions:

Suppose  $T$  is the smallest  $F$ -closed set.

If, for each set  $U$ ,

from the assumption “ $P(u)$  holds for every  $u \in U$ ”

we can show “ $P(v)$  holds for any  $v \in F(U)$ ,”

then  $P(t)$  holds for all  $t \in T$ .

Why?

## Structural Induction

The principle of structural induction on terms can also be re-stated using generating functions:

Suppose  $T$  is the smallest  $F$ -closed set.

If, for each set  $U$ ,

from the assumption “ $P(u)$  holds for every  $u \in U$ ”

we can show “ $P(v)$  holds for any  $v \in F(U)$ ,”

then  $P(t)$  holds for all  $t \in T$ .

Why? Because we assumed that  $T$  was the **smallest**  $F$ -closed set, i.e., that  $T \subseteq O$  for any other  $F$ -closed set  $O$ . But showing

for each set  $U$ ,

given  $P(u)$  for all  $u \in U$

we can show  $P(v)$  for all  $v \in F(U)$

amounts to showing that  $O =$  “the set of all terms satisfying  $P$ ” is itself an  $F$ -closed set, i.e.  $T \subseteq O$ , i.e., every element of  $T$  satisfies  $P$ .

## Structural Induction

Compare this with the structural induction principle for terms from last lecture:

If, for each term  $s$ ,

given  $P(r)$  for all immediate subterms  $r$  of  $s$

we can show  $P(s)$ ,

then  $P(t)$  holds for all  $t$ .

## Operational Semantics

## Abstract Machines

An **abstract machine** consists of:

- ◆ a set of **states**
- ◆ a **transition relation** on states, written  $\rightarrow$

## Operational semantics for Booleans

Syntax of terms and values

$t ::=$   
true  
false  
if  $t$  then  $t$  else  $t$

terms  
constant true  
constant false  
conditional

$v ::=$   
true  
false

values  
true value  
false value

The evaluation relation  $t \rightarrow t'$  is the smallest relation closed under the following rules:

if true then  $t_2$  else  $t_3 \rightarrow t_2$  (E-IFTRUE)

if false then  $t_2$  else  $t_3 \rightarrow t_3$  (E-IFFALSE)

$$\frac{t_1 \rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3} \quad (\text{E-IF})$$

## 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?

## Evaluation, more explicitly

$\rightarrow$  is the smallest two-place relation closed under the following rules:

$$((\text{if true then } t_2 \text{ else } t_3), t_2) \in \rightarrow$$

$$((\text{if false then } t_2 \text{ else } t_3), t_3) \in \rightarrow$$

$$(t_1, t'_1) \in \rightarrow$$

$$\frac{}{((\text{if } t_1 \text{ then } t_2 \text{ else } t_3), (\text{if } t'_1 \text{ then } t_2 \text{ else } t_3)) \in \rightarrow}$$

## Even more explicitly...

What is the generating function corresponding to these rules?

(exercise)

## Reasoning about Evaluation

## Derivations

We can record the “justification” for a particular pair of terms that are in the evaluation relation in the form of a tree.

(on the board)

Terminology:

- ◆ These trees are called **derivation trees** (or just **derivations**)
- ◆ The final statement in a derivation is its **conclusion**
- ◆ We say that the derivation is a **witness** for its conclusion (or a **proof** of its conclusion) — it records all the reasoning steps that justify the conclusion.

## Observation

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

1. the final rule used in  $\mathcal{D}$  is E-IFTRUE and we have  $t = \text{if true then } t_2 \text{ else } t_3$  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  $t = \text{if false then } t_2 \text{ else } t_3$  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  $t = \text{if } t_1 \text{ then } t_2 \text{ else } t_3$  and  $t' = \text{if } t'_1 \text{ then } t_2 \text{ else } t_3$ , for some  $t_1, t'_1, t_2$ , and  $t_3$ ; moreover, the immediate subderivation of  $\mathcal{D}$  witnesses  $(t_1, t'_1) \in \rightarrow$ .

## Induction on Derivations

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

Given an arbitrary derivation  $\mathcal{D}$  with conclusion  $t \rightarrow t'$ , 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 Derivations — Example

**Theorem:** If  $t \rightarrow t'$  — i.e., if  $(t, t') \in \rightarrow$  — then  $\text{size}(t) > \text{size}(t')$ .

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

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

## 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”) if there is no  $t'$  such that  $t \rightarrow t'$ .

A normal form is a state where the abstract machine is halted — i.e., it can be regarded as a “result” of evaluation.

## 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”) if there is no  $t'$  such that  $t \rightarrow t'$ .

A normal form is a state where the abstract machine is halted — i.e., it can be regarded as a “result” of evaluation.

Recall that we intended the set of **values** (the boolean constants **true** and **false**) to be exactly the possible “results of evaluation.”

Did we get this definition right?



## Values = normal forms

**Theorem:** A term  $t$  is a value iff it is in normal form.

**Proof:**

## Values = normal forms

**Theorem:** A term  $t$  is a value iff it is in normal form.

**Proof:** The  $\Rightarrow$  direction is immediate from the definition of the evaluation relation.

## Values = normal forms

**Theorem:** A term  $t$  is a value iff it is in normal form.

**Proof:** The  $\Rightarrow$  direction is immediate from the definition of the evaluation relation.

For the  $\Leftarrow$  direction,

## Values = normal forms

**Theorem:** A term  $t$  is a value iff it is in normal form.

**Proof:** The  $\Rightarrow$  direction is immediate from the definition of the evaluation relation.

For the  $\Leftarrow$  direction, it is convenient to prove the contrapositive: If  $t$  is **not** a value, then it is **not** a normal form.

## Values = normal forms

**Theorem:** A term  $t$  is a value iff it is in normal form.

**Proof:** The  $\Rightarrow$  direction is immediate from the definition of the evaluation relation.

For the  $\Leftarrow$  direction, it is convenient to prove the contrapositive: If  $t$  is **not** a value, then it is **not** a normal form. The argument goes by induction on  $t$ .

Note, first, that  $t$  must have the form `if  $t_1$  then  $t_2$  else  $t_3$`  (otherwise it would be a value). If  $t_1$  is `true` or `false`, then rule E-IFTRUE or E-IFFALSE applies to  $t$ , and we are done. Otherwise,  $t_1$  is not a value and so, by the induction hypothesis, there is some  $t'_1$  such that  $t_1 \rightarrow t'_1$ . But then rule E-IF yields

`if  $t_1$  then  $t_2$  else  $t_3 \rightarrow$  if  $t'_1$  then  $t_2$  else  $t_3$`

i.e.,  $t$  is not in normal form.

## Numbers

New syntactic forms

$t ::= \dots$	terms
0	constant zero
succ $t$	successor
pred $t$	predecessor
iszero $t$	zero test
$v ::= \dots$	values
nv	numeric value
$nv ::=$	numeric values
0	zero value
succ $nv$	successor value

New evaluation rules

$t \rightarrow t'$

$$\frac{t_1 \rightarrow t'_1}{\text{succ } t_1 \rightarrow \text{succ } t'_1} \quad (\text{E-SUCC})$$

$$\text{pred } 0 \rightarrow 0 \quad (\text{E-PREDZERO})$$

$$\text{pred } (\text{succ } nv_1) \rightarrow nv_1 \quad (\text{E-PREDSUCC})$$

$$\frac{t_1 \rightarrow t'_1}{\text{pred } t_1 \rightarrow \text{pred } t'_1} \quad (\text{E-PRED})$$

$$\text{iszero } 0 \rightarrow \text{true} \quad (\text{E-ISZEROZERO})$$

$$\text{iszero } (\text{succ } nv_1) \rightarrow \text{false} \quad (\text{E-ISZEROSUCC})$$

$$\frac{t_1 \rightarrow t'_1}{\text{iszero } t_1 \rightarrow \text{iszero } t'_1} \quad (\text{E-ISZERO})$$

## Values are normal forms

Our observation a few slides ago that all values are in normal form still holds for the extended language.

## Stuck terms

Is the converse true? I.e., is every normal form a value?

No: some terms are **stuck**.

Formally, a stuck term is one that is a normal form but not a value.

Stuck terms model run-time errors.

## Multi-step evaluation.

The **multi-step evaluation** relation,  $\longrightarrow^*$ , is the reflexive, transitive closure of single-step evaluation.

I.e., it is the smallest relation closed under the following rules:

$$\frac{t \longrightarrow t'}{t \longrightarrow^* t'}$$

$$t \longrightarrow^* t$$

$$\frac{t \longrightarrow^* t' \quad t' \longrightarrow^* t''}{t \longrightarrow^* t''}$$

## 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  $t \rightarrow^* t'$ .

**Proof:**

- ◆ First, recall that single-step evaluation strictly reduces the size of the term:  
if  $t \rightarrow t'$ , then  $\text{size}(t) > \text{size}(t')$
- ◆ Now, assume (for a contradiction) that  
 $t_0, t_1, t_2, t_3, t_4, \dots$   
is an infinite-length sequence such that  
 $t_0 \rightarrow t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow \dots$ ,
- ◆ Then  
 $\text{size}(t_0), \text{size}(t_1), \text{size}(t_2), \text{size}(t_3), \text{size}(t_4), \dots$   
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  
 $w_0 > w_1 > w_2 > \dots$  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  $W$ .