

## Comp 411 Notes - 21 Jan 2005

### SEMANTICS FOR A SIMPLE PROGRAMMING LANGUAGE, CONTINUED

Reproduced by Dan Smith  
(dlsmith@rice.edu)

## 1 Important Points from Previous Lecture

- On the meaning of derivation trees: can they be constructed in an infinite manner? In some contexts, yes. But we will only concern ourselves with finite derivation trees. The Winskel text (listed on the course Web site) has a good treatment of the formal meaning of inference rules and derivation trees.
- Four important rules hold in general for big step semantics:
  - Determinism (the big-step evaluation relation is a function)
  - Idempotence and identity on values ( $f(f(e)) = f(e)$  and  $f(v) = v$ )
  - All results are values
  - Classification (a loose term for type) of a value is preserved through operations on that value.

## 2 Small-Step Semantics for a Simple Programming Language

As before, the expressions are defined as follows:

$$e \in E ::= \text{T|F|if } e \text{ then } e \text{ else } e| \\ 0|e+|e-|\text{iszero } e$$

and values are:

$$v \in V ::= \text{T|F|0|}v+$$

The small-step semantics of a language describes the smallest amount of work possible to be performed on a program; usually, it is described as a relation on  $E \times E$ , a transformation on expressions.

The rules for this language are as follows:

$$\frac{e_1 \mapsto e'_1}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \mapsto \text{if } e'_1 \text{ then } e_2 \text{ else } e_3}$$

$$\frac{}{\text{if T then } e_1 \text{ else } e_2 \mapsto e_1} \qquad \frac{}{\text{if F then } e_1 \text{ else } e_2 \mapsto e_2}$$

$$\frac{e \mapsto e'}{e+ \mapsto e'+} \qquad \frac{e \mapsto e'}{e- \mapsto e'-} \qquad \frac{}{v+- \mapsto v}$$

$$\frac{e \mapsto e'}{\text{iszero } e \mapsto \text{iszero } e'} \qquad \frac{}{\text{iszero } 0 \mapsto \text{T}} \qquad \frac{}{\text{iszero } v+ \mapsto \text{F}}$$

Discussion:

- The following rules, if taken together, are incorrect because they overlap (i.e., they introduce nondeterminism into the  $\mapsto$  relation):

$$\frac{e \mapsto e'}{e- \mapsto e'-} \quad \frac{}{e+- \mapsto e}$$

- This set of rules satisfies the following properties:

- Determinism
- $v \not\mapsto$
- $e \mapsto^* v_1 \wedge e \mapsto^* v_2 \Rightarrow v_1 = v_2$

- The set of all programs can be subdivided into three non-overlapping subsets:  $V$ , the set of values;  $W$ , the “workable” set of programs (those for which  $\mapsto$  is defined); and  $S$ , the “stuck” programs (non-values for which  $\mapsto$  is undefined). We can describe all valid small-step transformations as one of  $W \mapsto W$ ,  $W \mapsto S$ , or  $W \mapsto V$ . An optimizing interpreter might also perform the transformations  $S \mapsto S$  and  $V \mapsto V$ . Any other transformation (such as  $S \mapsto V$ ) would be a violation of the semantics. (As a side note, type-safe languages are characterized by the fact that the set  $S$  is empty.)

### 3 Relating Small-Step and Big-Step Semantics

While the derivations look quite different, big-step and small-step semantics are closely related by the following property:

$$e \hookrightarrow v \Leftrightarrow e \mapsto^* v$$

## 4 Homework

Due with next problem set: Prove that  $e \hookrightarrow v \Leftrightarrow e \mapsto^* v$  for the language described in this and the previous lecture.