

50003 - Models of Computation - (Dr Raad) Lecture 3

Oliver Killane

24/10/21

Syntax of While

We can define a simple **While** language (if, else, while loops) to build programs from & to analyse.

$$\begin{aligned} B \in Bool & ::= true | false | E = E | E < E | B \& B | \neg B \dots \\ E \in Exp & ::= x | n | E + E | E \times E | \dots \\ C \in Com & ::= x := E | if\ B\ then\ C\ else\ C' | C; C' | skip | while\ B\ do\ C \end{aligned}$$

Where $x \in Var$ ranges over variable identifiers, and $n \in \mathbb{N}$ ranges over natural numbers.

States

A **state** is a partial function from variables to numbers. For state s , and variable x , $s(x)$ is defined, e.g:

$$s = (x \mapsto 2, y \mapsto 200, z \mapsto 20)$$

(In the current state, $x = 2, y = 200, z = 20$).

Partial Functions

A partial function is a mapping of every member of its domain, to at most one member of its codomain.

A state is a partial function as it is only defined for some variables.

For example:

$$\begin{aligned} s[x \mapsto 7](u) &= 7 && \text{if } u = x \\ &= s(u) && \text{otherwise} \end{aligned}$$

The small step semantics of **While** are defined using **configurations** of form:

$$\langle E, s \rangle, \langle B, s \rangle, \langle C, s \rangle$$

(Evaluating E , B , or C with respect to state s)

We can create a new state, where variable x equals value a , from an existing state s :

$$s'(u) \triangleq \alpha(x) = \begin{cases} a & u = x \\ s(u) & \text{otherwise} \end{cases}$$

$$s' = s[x \mapsto a] \text{ is equivalent to } \text{dom}(s') = \text{dom}(s) \wedge \forall y. [y \neq x \rightarrow s(y) = s'(y) \wedge s'(x) = a]$$

(s' equals s where x maps to a)

Expressions

$$\begin{aligned}
& \text{(W-EXP.LEFT)} \frac{\langle E_1, s \rangle \rightarrow_e \langle E'_1, s' \rangle}{\langle E_1 + E_2, s \rangle \rightarrow_e \langle E'_1 + E_2, s' \rangle} \\
& \text{(W-EXP.RIGHT)} \frac{\langle E, s \rangle \rightarrow_e \langle E', s' \rangle}{\langle n + E, s \rangle \rightarrow_e \langle n + E', s' \rangle} \\
& \text{(W-EXP.VAR)} \frac{}{\langle x, s \rangle \rightarrow_e \langle n, s \rangle} s(x) = n \\
& \text{(W-EXP.ADD)} \frac{}{\langle n_1 + n_2, s \rangle} \langle n_3, s \rangle n_3 = n_1 + n_2
\end{aligned}$$

These rules allow for side effects, despite the While language being side effect free in expression evaluation. We show this by changing state $s \rightarrow_e s'$.

We can show inductively (from the base cases W-EXP.VAR and W-EXP.ADD) that expression evaluation is side effect free.

Booleans

(Based on expressions, one can create the same for booleans) ($b \in \{true, false\}$)

AND

$$\begin{aligned}
& \text{(W-BOOL.AND.LEFT)} \frac{\langle B_1, s \rangle \rightarrow_b \langle B'_1, s' \rangle}{\langle B_1 \& B_2, s \rangle \rightarrow_b \langle B'_1 \& B_2, s' \rangle} \\
& \text{(W-BOOL.AND.RIGHT)} \frac{\langle B, s \rangle \rightarrow_b \langle B', s' \rangle}{\langle b \& B_2, s \rangle \rightarrow_b \langle b \& B', s' \rangle} \\
& \text{(W-BOOL.AND.TRUE)} \frac{}{\langle true \& true, s \rangle \rightarrow_b \langle true, s \rangle} \\
& \text{(W-BOOL.AND.FALSE)} \frac{}{\langle false \& b, s \rangle \rightarrow_b \langle true, s \rangle}
\end{aligned}$$

(Notice we do not short circuit, as the right arm may change the state. In a side effect free language, we could.)

EQUAL

$$\begin{aligned}
& \text{(W-BOOL.EQUAL.LEFT)} \frac{\langle E_1, s \rangle \rightarrow_e \langle E'_1, s' \rangle}{\langle E_1 = E_2, s \rangle \rightarrow_b \langle E'_1 = E_2, s' \rangle} \\
& \text{(W-BOOL.EQUAL.RIGHT)} \frac{\langle E, s \rangle \rightarrow_e \langle E', s' \rangle}{\langle n = E, s \rangle \rightarrow_b \langle n = E', s' \rangle} \\
& \text{(W-BOOL.EQUAL.TRUE)} \frac{}{\langle n_1 = n_2, s \rangle \rightarrow_b \langle true, s \rangle} n_1 = n_2 \\
& \text{(W-BOOL.EQUAL.FALSE)} \frac{}{\langle n_1 = n_2, s \rangle \rightarrow_b \langle false, s \rangle} n_1 \neq n_2
\end{aligned}$$

LESS

$$(W\text{-BOOL.LESS.LEFT}) \frac{\langle E_1, s \rangle \rightarrow_e \langle E'_1, s' \rangle}{\langle E_1 < E_2, s \rangle \rightarrow_b \langle E'_1 < E_2, s' \rangle}$$

$$(W\text{-BOOL.LESS.RIGHT}) \frac{\langle E, s \rangle \rightarrow_e \langle E', s' \rangle}{\langle n < E, s \rangle \rightarrow_b \langle n < E', s' \rangle}$$

$$(W\text{-BOOL.LESS.TRUE}) \frac{}{\langle n_1 < n_2, s \rangle \rightarrow_b \langle true, s \rangle} n_1 < n_2$$

$$(W\text{-BOOL.EQUAL.FALSE}) \frac{}{\langle n_1 < n_2, s \rangle \rightarrow_b \langle false, s \rangle} n_1 \geq n_2$$

NOT

$$(W\text{-BOOL.NOT}) \frac{}{\langle \neg true, s \rangle \rightarrow_b \langle false, s \rangle}$$

$$(W\text{-BOOL.NOT}) \frac{}{\langle \neg false, s \rangle \rightarrow_b \langle true, s \rangle}$$