Homework 2: Operational Semantics for WHILE

CS 252: Advanced Programming Languages
Student Ji An Lee
San José State University

1 MY ASSIGNMENT

Part 1: Rewrite the operational semantic rules for WHILE in LaTeX to use big-step operational semantics instead. Submit both your LaTeX source and the generated PDF file.

Extend your semantics with features to handle boolean values. **Do not treat these a binary operators.** Specifically, add support for:

- `and`
- `or`
- `not`

The exact behavior of these new features is up to you, but should seem reasonable to most programmers.

Part 2: Once you have your semantics defined, download `WhileInterp.hs` and implement the `evaluate` function, as well as any additional functions you need. Your implementation must be consistent with your operational semantics, *including your extensions for* `and`*,* `or`*, and* `not`. Also, you may not change any type signatures provided in the file.

Finally, implement the interpreter to match your semantics.

Zip all files together into `hw2.zip` and submit to Canvas.

$$
\begin{array}{lll}
e ::= & & \textit{Expressions} \\
& x & \text{variables/addresses} \\
& v & \text{values} \\
& x := e & \text{assignment} \\
& e; e & \text{sequential expressions} \\
& e \mathrel{op} e & \text{binary operations} \\
& \texttt{if}\ e\ \texttt{then}\ e\ \texttt{else}\ e & \text{conditional expressions} \\
& \texttt{while}\ (e)\ e & \text{while expressions} \\
\\
v ::= & & \textit{Values} \\
& i & \text{integer values} \\
& b & \text{boolean values} \\
\\
op ::= & + \mid - \mid * \mid / \mid > \mid >= \mid < \mid <= & \textit{Binary operators}
\end{array}
$$

Figure 1: The WHILE language

Evaluation Rules: $\boxed{(e_0, \sigma_0) \Downarrow (e_1, \sigma_1)}$

$$[\textsc{bs-value}] \qquad \frac{}{(v, \sigma) \Downarrow (v, \sigma)}$$

$$[\textsc{bs-var}] \qquad \frac{v = \sigma_0(x_0)}{(x_0, \sigma_0) \Downarrow (v, \sigma_0)}$$

$$[\textsc{bs-assign}] \qquad \frac{(e_0, \sigma_0) \Downarrow (v_0, \sigma_1) \quad \textit{where}\ \sigma_2 = \sigma_1(x \mapsto v_0)}{(x := e_0, \sigma_0) \Downarrow (v_0, \sigma_2)}$$

$$[\textsc{bs-seq}] \qquad \frac{(e_0, \sigma_0) \Downarrow (v_0, \sigma_1) \quad (e_1, \sigma_1) \Downarrow (v_1, \sigma_2)}{(e_0; e_1, \sigma_0) \Downarrow (v_1, \sigma_2)}$$

$$[\textsc{bs-op}] \qquad \frac{(e_0, \sigma_0) \Downarrow (n_0, \sigma_1) \quad (e_1, \sigma_1) \Downarrow (n_1, \sigma_2) \quad n_2 = n_0 \mathrel{op} n_1}{(e_0 \mathrel{op} e_1, \sigma_0) \Downarrow (n_2, \sigma_2)}$$

$$[\textsc{bs-iftrue}] \qquad \frac{(e_0, \sigma_0) \Downarrow (\mathit{True}, \sigma_1) \quad (e_1, \sigma_1) \Downarrow (v_0, \sigma_2)}{(\texttt{if}\ \texttt{e}_0\ \texttt{e}_1\ \texttt{e}_2, \sigma_0) \Downarrow (v_0, \sigma_2)}$$

$$[\textsc{bs-iffalse}] \qquad \frac{(e_0, \sigma_0) \Downarrow (\mathit{False}, \sigma_1) \quad (e_2, \sigma_1) \Downarrow (v_0, \sigma_2)}{(\texttt{if}\ \texttt{e}_0\ \texttt{e}_1\ \texttt{e}_2, \sigma_0) \Downarrow (v_0, \sigma_2)}$$

$$[\textsc{bs-while-False}] \qquad \frac{(e_0, \sigma_0) \Downarrow (\mathit{False}, \sigma_1)}{(\mathit{while}(e_0)e_1, \sigma_0) \Downarrow (\mathit{False}, \sigma_1)}$$

$$[\textsc{bs-while-True}] \qquad \frac{(e_0, \sigma_0) \Downarrow (\mathit{True}, \sigma_1) \quad (e_1, \sigma_1) \Downarrow (v_0, \sigma_2) \quad (\mathit{while}(e_0)e_1, \sigma_2) \Downarrow (v_1, \sigma_3)}{(\mathit{while}(e_0)e_1, \sigma_0) \Downarrow (v_1, \sigma_3)}$$

Figure 2: Big-step semantics for WHILE

Continued..

$$[\textsc{bs-and-true}] \qquad \frac{(e_0, \sigma_0) \Downarrow (True, \sigma_1) \quad (e_1, \sigma_1) \Downarrow (b, \sigma_2) \quad | \;\; b \in bool}{(AND\ e_0\ e_1, \sigma_0) \Downarrow (b, \sigma_2)}$$

$$[\textsc{bs-and-false}] \qquad \frac{(e_0, \sigma_0) \Downarrow (False, \sigma_1)}{(AND\ e_0\ e_1, \sigma_0) \Downarrow (False, \sigma_1)}$$

$$[\textsc{bs-or-true}] \qquad \frac{(e_0, \sigma_0) \Downarrow (True, \sigma_1)}{(OR\ e_0\ e_1, \sigma_0) \Downarrow (True, \sigma_1)}$$

$$[\textsc{bs-or-false}] \qquad \frac{(e_0, \sigma_0) \Downarrow (False, \sigma_1) \quad (e_1, \sigma_1) \Downarrow (b, \sigma_2) \quad | \;\; b \in bool}{(OR\ e_0\ e_1, \sigma_0) \Downarrow (b, \sigma_2)}$$

$$[\textsc{bs-not-true}] \qquad \frac{(e_0, \sigma_0) \Downarrow (True, \sigma_1)}{(NOT\ e_0, \sigma_0) \Downarrow (False, \sigma_1)}$$

$$[\textsc{bs-not-false}] \qquad \frac{(e_0, \sigma_0) \Downarrow (False, \sigma_1)}{(NOT\ e_0, \sigma_0) \Downarrow (True, \sigma_1)}$$

Figure 3: Big-step semantics for WHILE