

CS 565 Spring 2022 Homework 3 (Big-Step Semantics)

Your name: Shuang Wu (wu1716@purdue.edu)

February 15, 2022

Using the big-step operational semantics of IMP (shown in Figure 1), either (i) fill in the final states produced by each of the following IMP programs and give a derivation justifying your answer **or** (ii) state that the program does not terminate. **Note:** You don't need to supply the derivations for any arithmetic or boolean expressions, just for the IMP statements.

Problem 1 (1 point).

$$\frac{\text{E-CONST} \frac{}{\emptyset, 0 \Downarrow_A 0} \quad \frac{\text{E-CONST} \frac{}{[X \mapsto 0], 1 \Downarrow_A 1}}{\text{E-Ass} \frac{\emptyset, X := 0 \Downarrow [X \mapsto 0] \quad [X \mapsto 0], Y := 1 \Downarrow [X \mapsto 0][Y \mapsto 1]}{\emptyset, X := 0; Y := 1 \Downarrow [X \mapsto 0][Y \mapsto 1]} \text{E-Ass} \quad \text{E-SEQ}}$$

Problem 2 (1 point).

$$\frac{\text{E-VAR} \frac{}{[X \mapsto 2], X \Downarrow_A 2} \quad \frac{\text{E-CONST} \frac{}{1 < 2}}{\text{E-LEF} \frac{[X \mapsto 2], X \leq 1 \Downarrow_B \text{false} \quad [X \mapsto 2], 1 \Downarrow_A 1}}{\text{E-Ass} \frac{[X \mapsto 2], Z := 4 \Downarrow [X \mapsto 2][Z \mapsto 4] \quad \frac{\text{E-COUNT} \frac{}{[X \mapsto 2], 4 \Downarrow_A 4}}{\text{E-Ass} \frac{[X \mapsto 2], Z := 4 \Downarrow [X \mapsto 2][Z \mapsto 4]}{\text{E-IFFALSE} \frac{[X \mapsto 2], \text{if } (X \leq 1) \text{ then } Y := X + 3 \text{ else } Z := 4 \text{ end } \Downarrow [X \mapsto 2][Z \mapsto 4]}}}}$$

Problem 3 (1 point).

$[X \mapsto 0], \mathbf{while} (X \leq 1) \mathbf{do} Y := Y + 1 \mathbf{end} \Downarrow$

The program does not terminate.

Problem 4 (1 point).

$$\begin{array}{c}
\text{E-VAR} \frac{}{[Y \mapsto 0], Y \Downarrow_A 0} \quad \frac{}{[Y \mapsto 0], 1 \Downarrow_A 1} \text{E-CONST} \quad \frac{}{0 \leq 1} \\
\text{E-LET} \frac{}{[Y \mapsto 0], Y \leq 1 \Downarrow_B \text{true}} \\
\text{E-VAR} \frac{}{[Y \mapsto 0], Y \Downarrow_A 0} \quad \frac{}{[Y \mapsto 0], 1 \Downarrow_A 1} \text{E-CONST} \\
\frac{}{[Y \mapsto 0], Y + 1 \Downarrow_A 1} \text{E-ADD} \\
\frac{}{[Y \mapsto 0], Y := Y + 1 \Downarrow [Y \mapsto 0][Y \mapsto 1]} \text{E-Ass} \\
\text{E-VAR} \frac{}{[Y \mapsto 0][Y \mapsto 1], Y \Downarrow_A 1} \quad \frac{}{[Y \mapsto 0][Y \mapsto 1], 1 \Downarrow_A 1} \text{E-CONST} \quad \frac{}{1 \leq 1} \\
\text{E-LET} \frac{}{[Y \mapsto 0][Y \mapsto 1], (Y \leq 1) \Downarrow_B \text{true}} \\
\text{E-VAR} \frac{}{[Y \mapsto 0][Y \mapsto 1], Y \Downarrow_A 1} \quad \frac{}{[Y \mapsto 0][Y \mapsto 1], 1 \Downarrow_A 1} \text{E-CONST} \\
\frac{}{[Y \mapsto 0][Y \mapsto 1], Y := Y + 1 \Downarrow_A 2} \text{E-ADD} \\
\frac{}{[Y \mapsto 0][Y \mapsto 1], Y := Y + 1 \Downarrow [Y \mapsto 0][Y \mapsto 1][Y \mapsto 2]} \text{E-Ass} \\
\text{E-VAR} \frac{}{[Y \mapsto 0][Y \mapsto 1][Y \mapsto 2], Y \Downarrow_A 2} \quad \frac{}{[Y \mapsto 0][Y \mapsto 1][Y \mapsto 2], 1 \Downarrow_A 1} \text{E-CONST} \quad \frac{}{1 < 2} \\
\text{E-LEF} \frac{}{[Y \mapsto 0][Y \mapsto 1][Y \mapsto 2], Y \leq 1 \Downarrow_B \text{false}} \\
\frac{}{[Y \mapsto 0][Y \mapsto 1][Y \mapsto 2], \mathbf{while} (Y \leq 1) \mathbf{do} Y := Y + 1 \mathbf{end} \Downarrow [Y \mapsto 0][Y \mapsto 1][Y \mapsto 2]} \text{E-WHILEFALSE} \\
\frac{}{[Y \mapsto 0][Y \mapsto 1], \mathbf{while} (Y \leq 1) \mathbf{do} Y := Y + 1 \mathbf{end} \Downarrow [Y \mapsto 0][Y \mapsto 1][Y \mapsto 2]} \text{E-WHILETRUE} \\
\frac{}{[Y \mapsto 0], \mathbf{while} (Y \leq 1) \mathbf{do} Y := Y + 1 \mathbf{end} \Downarrow [Y \mapsto 0][Y \mapsto 1][Y \mapsto 2]} \text{E-WHILETRUE}
\end{array}$$

Problem 5 (2 points). Several cutting edge languages, including Perl, Visual Basic, and Pascal include a **repeat c until b** loop construct. These loops differ from the standard **while** loops in two ways:

1. the loop guard is checked /after/ the execution of the body, so the loop always executes at least once.
2. the loop continues executing as long as the condition is false.

Write down the big-step reduction rules for **repeat** loops in IMP.

$$\frac{\sigma, c \Downarrow \sigma_1 \quad \sigma_1, b \Downarrow_B \text{true}}{\sigma, \text{repeat } c \text{ until } b \Downarrow \sigma_1} \text{E-REPEATTRUE}$$

$$\frac{\sigma, c \Downarrow \sigma_1 \quad \sigma_1, b \Downarrow_B \text{false} \quad \sigma_1, \text{repeat } c \text{ until } b \Downarrow \sigma_2}{\sigma, \text{repeat } c \text{ until } b \Downarrow \sigma_2} \text{E-REPEATFALSE}$$

$\frac{}{\sigma, x \Downarrow_A \sigma(x)} \text{ (E-VAR)}$	$\frac{}{\sigma, n \Downarrow_A n} \text{ (E-CONST)}$	$\frac{\sigma, a_1 \Downarrow_A m \quad a_2 \Downarrow_A n}{\sigma, a_1 + a_2 \Downarrow_A m + n} \text{ (E-ADD)}$
$\frac{}{\sigma, \text{true} \Downarrow_B \text{true}} \text{ (E-TRUE)}$	$\frac{}{\sigma, \text{false} \Downarrow_B \text{false}} \text{ (E-FALSE)}$	$\frac{\sigma, b_1 \Downarrow_B t \quad b_2 \Downarrow_B v}{\sigma, b_1 \wedge b_2 \Downarrow_B t \wedge v} \text{ (E-AND)}$
$\frac{\sigma, b \Downarrow_B \text{true}}{\sigma, \neg b \Downarrow_B \text{false}} \text{ (E-NOTT)}$	$\frac{\sigma, b \Downarrow_B \text{false}}{\sigma, \neg b \Downarrow_B \text{true}} \text{ (E-NOTF)}$	
$\frac{\sigma, a_1 \Downarrow_A n \quad \sigma, a_2 \Downarrow_A n}{\sigma, a_1 = a_2 \Downarrow_B \text{true}} \text{ (E-CMPT)}$	$\frac{\sigma, a_1 \Downarrow_A m \quad \sigma, a_2 \Downarrow_A n \quad m \neq n}{\sigma, a_1 = a_2 \Downarrow_B \text{false}} \text{ (E-CMPF)}$	
$\frac{\sigma, a_1 \Downarrow_A m \quad \sigma, a_2 \Downarrow_A n \quad m \leq n}{\sigma, a_1 \leq a_2 \Downarrow_B \text{true}} \text{ (E-LET)}$	$\frac{\sigma, a_1 \Downarrow_A m \quad \sigma, a_2 \Downarrow_A n \quad n < m}{\sigma, a_1 \leq a_2 \Downarrow_B \text{false}} \text{ (E-LEF)}$	
$\frac{}{\sigma, \text{skip} \Downarrow \sigma} \text{ (E-SKIP)}$	$\frac{\sigma, a \Downarrow_A n}{\sigma, x := a \Downarrow \sigma[x \mapsto n]} \text{ (E-ASS)}$	
$\frac{\sigma, c_1 \Downarrow \sigma_1 \quad \sigma_1, c_2 \Downarrow \sigma_2}{\sigma, c_1; c_2 \Downarrow \sigma_2} \text{ (E-SEQ)}$	$\frac{\sigma, b \Downarrow_B \text{true} \quad \sigma, c_1 \Downarrow \sigma_1}{\sigma, \text{if } b \text{ then } c_1 \text{ else } c_2 \Downarrow \sigma_1} \text{ (E-IFTRUE)}$	
$\frac{\sigma, b \Downarrow_B \text{false} \quad \sigma, c_2 \Downarrow \sigma_2}{\sigma, \text{if } b \text{ then } c_1 \text{ else } c_2 \Downarrow \sigma_2} \text{ (E-IFFALSE)}$	$\frac{\sigma, b \Downarrow_B \text{false}}{\sigma, \text{while } b \text{ do } c \Downarrow \sigma} \text{ (E-WHILEFALSE)}$	
$\frac{\sigma, b \Downarrow_B \text{true} \quad \sigma, c \Downarrow \sigma_1 \quad \sigma_1, \text{while } b \text{ do } c \Downarrow \sigma_2}{\sigma, \text{while } b \text{ do } c \Downarrow \sigma_2} \text{ (E-WHILETRUE)}$		

Figure 1: Big-step semantics of arithmetic expressions, boolean expressions, and regular IMP.