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

Your name: ____________________________________

February 5, 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).

$$\emptyset\ ,\ \mathsf{X} := 0;\ \mathsf{Y} := 1 \Downarrow$$

Problem 2 (1 point).

$$[\mathsf{X} \mapsto 2],\ \textbf{if}\ (\mathsf{X} \leq 1)\textbf{then}\ \mathsf{Y} := \mathsf{X} + 3\ \textbf{else}\ \mathsf{Z} := 4\ \mathsf{end} \Downarrow$$

Problem 3 (1 point).

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

Problem 4 (1 point).

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

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.

$$\frac{}{\sigma, \mathsf{x} \Downarrow_A \sigma(x)} \text{(E-Var)} \qquad \frac{}{\sigma, \mathsf{n} \Downarrow_A \mathsf{n}} \text{(E-Const)} \qquad \frac{\sigma, \mathsf{a}_1 \Downarrow_A \mathsf{m} \qquad \mathsf{a}_2 \Downarrow_A \mathsf{n}}{\sigma, \mathsf{a}_1 + \mathsf{a}_2 \Downarrow_A \mathsf{m} + \mathsf{n}} \text{(E-Add)}$$

$$\frac{}{\sigma, \mathsf{true} \Downarrow_B \mathsf{true}} \text{(E-True)} \qquad \frac{}{\sigma, \mathsf{false} \Downarrow_B \mathsf{false}} \text{(E-False)} \qquad \frac{\sigma, \mathsf{b}_1 \Downarrow_B \mathsf{t} \qquad \mathsf{b}_2 \Downarrow_B \mathsf{v}}{\sigma, \mathsf{b}_1 \wedge \mathsf{b}_2 \Downarrow_B \mathsf{t} \ \wedge \mathsf{v}} \text{(E-And)}$$

$$\frac{\sigma, \mathsf{b} \Downarrow_B \mathsf{true}}{\sigma, \neg\mathsf{b} \Downarrow_B \mathsf{false}} \text{(E-NotT)} \qquad \frac{\sigma, \mathsf{b} \Downarrow_B \mathsf{false}}{\sigma, \neg\mathsf{b} \Downarrow_B \mathsf{true}} \text{(E-NotF)}$$

$$\frac{\sigma, \mathsf{a}_1 \Downarrow_A \mathsf{n} \qquad \sigma, \mathsf{a}_2 \Downarrow_A \mathsf{n}}{\sigma, \mathsf{a}_1 = \mathsf{a}_2 \Downarrow_B \mathsf{true}} \text{(E-CmpT)} \qquad \frac{\sigma, \mathsf{a}_1 \Downarrow_A \mathsf{m} \qquad \sigma, \mathsf{a}_2 \Downarrow_A \mathsf{n} \qquad m \neq n}{\sigma, \mathsf{a}_1 = \mathsf{a}_2 \Downarrow_B \mathsf{true}} \text{(E-CmpF)}$$

$$\frac{}{\sigma, \textbf{skip} \Downarrow \sigma} \text{(E-Skip)} \qquad \frac{\sigma, \mathsf{a} \Downarrow_A \mathsf{n}}{\sigma, \mathsf{x} := \mathsf{a} \Downarrow \sigma[\mathsf{x} \mapsto \mathsf{n}]} \text{(E-Ass)}$$

$$\frac{\sigma, \mathsf{c}_1 \Downarrow \sigma_1 \qquad \sigma_1, \mathsf{c}_2 \Downarrow \sigma_2}{\sigma, \mathsf{c}_1; \mathsf{c}_2 \Downarrow \sigma_2} \text{(E-Seq)} \qquad \frac{\sigma, \mathsf{b} \Downarrow_B \mathsf{true} \qquad \sigma, \mathsf{c}_1 \Downarrow \sigma_1}{\sigma, \textbf{if } \mathsf{b} \textbf{ then } \mathsf{c}_1 \text{ else } \mathsf{c}_2 \Downarrow \sigma_1} \text{(E-IfTrue)}$$

$$\frac{\sigma, \mathsf{b} \Downarrow_B \mathsf{false} \qquad \sigma, \mathsf{c}_2 \Downarrow \sigma_2}{\sigma, \textbf{if } \mathsf{b} \textbf{ then } \mathsf{c}_1 \text{ else } \mathsf{c}_2 \Downarrow \sigma_2} \text{(E-IfFalse)} \qquad \frac{\sigma, \mathsf{b} \Downarrow_B \mathsf{false}}{\sigma, \textbf{while } \mathsf{b} \textbf{ do } \mathsf{c} \Downarrow \sigma} \text{(E-WhileFalse)}$$

$$\frac{\sigma, \mathsf{b} \Downarrow_B \mathsf{true} \qquad \sigma, \mathsf{c} \Downarrow \sigma_1 \qquad \sigma_1, \textbf{while } \mathsf{b} \textbf{ do } \mathsf{c} \Downarrow \sigma_2}{\sigma, \textbf{while } \mathsf{b} \textbf{ do } \mathsf{c} \Downarrow \sigma_2} \text{(E-WhileTrue)}$$

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