1 Instructions

This assignment is to be completed by hand (or in LaTeX if you know how to use it). Submit your answers as a PDF file via Gradescope. If you don’t have easy access to a scanner, you can use the one in SEO 1120, the main CS office – the staff will be happy to help you. As always, please don’t hesitate to ask for help on Piazza (https://piazza.com/class/jkh8q52qrh06v).

2 Operational Semantics of IMP

Here are the operational semantics rules for a simple imperative programming language, using the “hybrid style” of big steps for expressions and small steps for commands.

\[
\begin{align*}
(n & \text{ a number}) \quad (b & \text{ a boolean}) \quad (\sigma(x) = v) \\
(n, \sigma) & \Downarrow n & (b, \sigma) & \Downarrow b & (x, \sigma) & \Downarrow v \\
(e_1, \sigma) & \Downarrow v_1 & (e_2, \sigma) & \Downarrow v_2 & (v_1 \oplus v_2 = v) \\
(e_1 \oplus e_2, \sigma) & \Downarrow v & \text{where } \oplus & \text{ is an arithmetic or boolean operator} \\
(e, \sigma) & \Downarrow \text{true} & (e_1, \sigma) & \Downarrow v \\
(\text{if } e \text{ then } e_1 \text{ else } e_2, \sigma) & \Downarrow v & (e, \sigma) & \Downarrow \text{false} & (e_2, \sigma) & \Downarrow v \\
(x := e, \sigma) & \rightarrow (\text{skip}, \sigma[x \mapsto v]) & (c_1, \sigma) & \rightarrow (c_1', \sigma') & (c_1; c_2, \sigma) & \rightarrow (c_1'; c_2, \sigma') & (\text{skip}; c_2, \sigma) & \rightarrow (c_2, \sigma)
\end{align*}
\]

3 Problems

There are four problems in all. Each problem is on a separate page. Use as much space as you need for each problem. You can add extra pages if you need to.
1. (6 points) Using the rules above, construct a proof tree showing that \( (x + (2 \ast y), \{ x = 2, y = 3 \}) \downarrow 8 \).
In other words, show that \( x + (2 \ast y) \) evaluates to 8 in the state where \( x = 2 \) and \( y = 3 \).
2. (3 points) Construct a proof tree showing that

\[(z := x + (2 \ast y); \ x := \text{if } z = 7 \text{ then } 3 \text{ else } 4, \{x = 2, y = 3\}) \rightarrow \]
\[(\text{skip}; \ x := \text{if } z = 7 \text{ then } 3 \text{ else } 4, \{x = 2, y = 3, z = 8\})\]

You can write “P1” to stand for the proof tree from the previous problem.
3. (7 points) Construct a proof tree for the next step that 

\[(x := \text{if } z = 7 \text{ then } 3 \text{ else } 4, \{x = 2, y = 3, z = 8\})\]

takes.
4. (9 points) Suppose we extended the language with a command “c₁ andthen c₂ if e” that behaves as follows:

- First, it executes c₁ normally.
- If e is true in the resulting environment, it then executes c₂.
- Otherwise, it ignores c₂ and is finished executing.

In other words, to execute c₁ andthen c₂ if e, first execute c₁ normally, and then execute c₂ only if e is true.

Give small-step semantic rules for c₁ andthen c₂ if e. Remember that a command becomes “skip” when it is finished executing. As a test case, if you’ve written your rules correctly, x := 3 andthen y := 4 if x = 3 should step to (skip, {x = 3, y = 4}) in three small steps.

Hint: it is probably easiest to define the command using three separate rules.