CS 476 HW3 -- Proof Trees and Operational Semantics

Sandeep Joshi

TOTAL POINTS

22 / 25

QUESTION 1

1 Problem 1 6 / 6

✓ + 0 pts graded

✓ + 1 pts op rule for x + (2 * y)

✓ + 1 pts var rules

✓ + 1 pts num rule

✓ + 1 pts op rule for 2 * y

✓ + 1 pts correct values

✓ + 1 pts overall structure

- 1 pts Incorrect environment(s)

Bravo!

QUESTION 2

2 Problem 2 1 / 3

✓ + 0 pts graded

✓ + 1 pts structural rule for sequence

+ 1 pts assignment rule

- 1 pts evaluated second command in sequence

+ 1 pts overall structure

- 0.5 pts Extra rules

1 This needs to be expanded out, one step at a time. At this point, the left-hand side should be (z := x + (2*y), sigma).

QUESTION 3

3 Problem 3 7 / 7

✓ + 0 pts graded

✓ + 3 pts correct next step

✓ + 1 pts assignment rule

✓ + 1 pts if-then-else rule

✓ + 1 pts used false case

✓ + 1 pts leaf proofs

- 0.5 pts Incorrect env or other small mistake

QUESTION 4

4 Problem 4 8 / 9

✓ + 0 pts graded

✓ + 2 pts conclusion of each rule is a small step

✓ + 1 pts structural rule for c1 exists

✓ + 1 pts structural rule uses small-step above the line

✓ + 1 pts structural rule correctly steps c1 below the line

✓ + 1 pts true case steps to c2

✓ + 1 pts false case steps to skip

✓ + 2 pts true and false cases only apply once c1 has been reduced to skip

- 1 pts big-step rules

✓ - 1 pts extra rules

- 0.5 pts c2' in structural rule for c1

+ 3 pts used sequencing to handle c1

- 0.5 pts small steps for expressions

2 C1 won't necessarily reduce to a single assignment; for instance, it could be skip to begin with.

3 The LHS here should be "skip andthen C2 if e". Then the rule above would be unnecessary.

1. (6 points) Using the rules above, construct a proof tree showing that $(\mathrm{x} + (2 * \mathrm{y}), \{x = 2, y = 3\}) \Downarrow 8$. In other words, show that $\mathrm{x} + (2 * \mathrm{y})$ evaluates to 8 in the state where $x = 2$ and $y = 3$.

$$\textit{State} : \sigma = \{x = 2, y = 3\}$$

$$\dfrac{\dfrac{\{x = 2, y = 3\}(x) = 2}{(x, \{x = 2, y = 3\}) \Downarrow 2} \qquad \dfrac{\dfrac{}{(2, \{x = 2, y = 3\}) \Downarrow 2} \qquad \dfrac{\{x = 2, y = 3\}(y) = 3}{(y, \{x = 2, y = 3\}) \Downarrow 3}}{((2 * y), \{x = 2, y = 3\}) \Downarrow 6}}{(x + (2 * y), \{x = 2, y = 3\}) \Downarrow 8}$$

1 Problem 1 6 / 6

- ✓ + 0 pts graded
- ✓ + 1 pts op rule for x + (2 * y)
- ✓ + 1 pts var rules
- ✓ + 1 pts num rule
- ✓ + 1 pts op rule for 2 * y
- ✓ + 1 pts correct values
- ✓ + 1 pts overall structure
- - 1 pts Incorrect environment(s)

Bravo!

2. (3 points) Construct a proof tree showing that

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

You can write "P1" to stand for the proof tree from the previous problem.

$$\sigma = \{x = 2, y = 3\} \sigma' = \{x = 2, y = 3, z = 8\}$$

$$\frac{\dfrac{P1}{(z := 1\ \sigma) \rightarrow (skip, \sigma')}}{(z := x + (2 * y); x := \text{if } z = 7 \text{ then } 3 \text{ else } 4, \sigma) \rightarrow (\text{skip}; x := \text{if } z = 7 \text{ then } 3 \text{ else } 4, \sigma')}$$

2 Problem 2 1 / 3

✓ **+ 0 pts graded**

✓ **+ 1 pts structural rule for sequence**

+ 1 pts assignment rule

- 1 pts evaluated second command in sequence

+ 1 pts overall structure

- 0.5 pts Extra rules

1 This needs to be expanded out, one step at a time. At this point, the left-hand side should be (z := x + (2*y), sigma).

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.

$$\sigma = \{x = 2, y = 3, z = 8\}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\sigma(z) = 8}{(z, \sigma) \Downarrow 8} \quad \dfrac{7 \text{ is a number}}{(7, \sigma) \Downarrow 7}}{(z = 7, \sigma) \Downarrow \mathit{False}} \quad \dfrac{4 \text{ is a number}}{(4, \sigma) \Downarrow 4}}{(\text{if } z = 7 \text{ then } 3 \text{ else } 4, \sigma) \Downarrow 4}}{(x := 4, \{x = 2, y = 3, z = 8\}) \to (\mathit{skip}, \{x = 4, y = 3, z = 8\})}$$

3 Problem 3 7 / 7

- ✓ + 0 pts graded
- ✓ + 3 pts correct next step
- ✓ + 1 pts assignment rule
- ✓ + 1 pts if-then-else rule
- ✓ + 1 pts used false case
- ✓ + 1 pts leaf proofs
  - - 0.5 pts Incorrect env or other small mistake

4. (9 points) Suppose we extended the language with a command "c_1 andthen c_2 if e" that behaves as follows:

   - First, it executes c_1 normally.
   - If e is true in the resulting environment, it then executes c_2.
   - Otherwise, it ignores c_2 and is finished executing.

   In other words, to execute c_1 andthen c_2 if e, first execute c_1 normally, and then execute c_2 only if e is true.

   Give small-step semantic rules for c_1 andthen c_2 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 $(\text{skip}, \{x = 3, y = 4\})$ in three small steps.

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

$$\frac{(C_1, \sigma) \to (C_1', \sigma')}{(C_1 \text{ and then } C_2 \text{ if } e, \sigma) \to (C_1' \text{ and then } C_2 \text{ if } e, \sigma')}$$

$$\frac{(m, \sigma) \Downarrow v}{(x := m \text{ andthen if } e) \to (\text{skip } C_2 \text{ if } e, \sigma[x \mapsto v])}$$ 2

$$(\text{skip } C_2 \text{ if } e, \sigma) \to (C_2 \text{ if } e, \sigma)$$ 3

$$\frac{(e, \sigma) \Downarrow True \qquad (C_2, \sigma) \to (C_2', \sigma')}{(C_2 \text{ if } e, \sigma) \to (C_2', \sigma')}$$

$$\frac{(e, \sigma) \Downarrow False}{(C_2 \text{ if } e, \sigma) \to (skip, \sigma)}$$

4 Problem 4 8 / 9

- ✓ + 0 pts graded
- ✓ + 2 pts conclusion of each rule is a small step
- ✓ + 1 pts structural rule for c1 exists
- ✓ + 1 pts structural rule uses small-step above the line
- ✓ + 1 pts structural rule correctly steps c1 below the line
- ✓ + 1 pts true case steps to c2
- ✓ + 1 pts false case steps to skip
- ✓ + 2 pts true and false cases only apply once c1 has been reduced to skip
- - 1 pts big-step rules
- ✓ - 1 pts extra rules
- - 0.5 pts c2' in structural rule for c1
- + 3 pts used sequencing to handle c1
- - 0.5 pts small steps for expressions

2 C1 won't necessarily reduce to a single assignment; for instance, it could be skip to begin with.

3 The LHS here should be "skip andthen C2 if e". Then the rule above would be unnecessary.