

14F-1 Bookkeeping

- 0 pts Correct

Peer Review ID: 70906224 — enter this when you fill out your peer evaluation via gradescope

4F-2

The bug in the let rule is that $VC(c, B)$ is evaluated with x as e , but after the execution of "let $x=e$ in c ", the value in x is the previous value in x before the execution of the command.

Let x' be an unused variable name

$$\begin{aligned} & VC(\text{let } x=e \text{ in } c, B) \\ &= VC(x'=x; x=e; c; x=x', B) \\ &= VC(x'=x; x=e; c; [x'/x]B) \\ &= VC(x'=x; x=e; VC(c, [x'/x]B)) \\ &= VC(x'=x; [e/x]VC(c, [x'/x]B)) \\ &= [x/x'] [e/x]VC(c, [x'/x]B) \end{aligned}$$

2 4F-2 VCGen for Let

- 0 pts Correct

4F-3

1. $c = \text{let } x=6 \text{ in skip}$
2. post condition $B: x > 5$
3. State $\sigma: \sigma(x) = 1$
4.
$$\begin{aligned} \text{VC}(\text{let } x=6 \text{ in skip}, x > 5) &= x > 5 \\ &= [6/x] \text{VC}(\text{skip}, x > 5) \\ &= [6/x] x > 5 \\ &= 6 > 5 \\ &= \text{true} \end{aligned}$$

Thus, $\sigma \not\models \text{VC}(c, B)$

5. $\langle \text{skip}, \sigma[x=1] \rangle \Downarrow \sigma'[x=1]$
6. $\sigma' \not\models B$ because $1 > 5$ is false

3 4F-3 VCGen Mistakes

- 0 pts Correct

4F-4

The only difference between a do-while and a while-do loop is that in a do-while loop, the pre-condition must hold even after the execution of the loop body independent of the loop guard.

$$\vdash \{A\} c \{A\} \quad \vdash \{A \wedge b\} c \{A\}$$

$$\vdash \{A\} \text{ do } c \text{ while } b \{A \wedge \neg b\}$$

In the above rule, we have two hypothesis:

$$\vdash \{A\} c \{A\} \quad \text{and} \quad \vdash \{A \wedge b\} c \{A\}$$

The pre-condition of the first hypothesis is a more general version of the pre-condition of the second hypothesis.

$$\vdash \{A\} c \{A\} \Rightarrow \vdash \{A \wedge b\} c \{A\}.$$

Thus the rule can be written as:

$$\{A\} c \{A\}$$

$$\vdash \{A\} \text{ do } c \text{ while } b \{A \wedge \neg b\}$$

One limitation of the above rule is that it does not allow the pre and post conditions to be very different. If we want to reason about different conditions, we can use the following rule:

$$\vdash \{A\} c \{B\} \quad \vdash \{B \wedge b\} c \{B\}$$

$$\vdash \{A\} \text{ do } c \text{ while } b \{B \wedge \neg b\}$$

4 4F-4 Axiomatic Do-While

- 0 pts Correct