1 4F-1 Bookkeeping

- 0 pts Correct

Peer Review ID: 70873991 — enter this when you fill out your peer evaluation via gradescope

Exercise 4F-2:

That rule for let has a bug where the original value of the variable x is not restored. Since for the let command, this new value e of x has a scope only in the c command.

A correct rule for let needs to restore the original value of x, which we store at a temporary variable named temp.

$VC(\text{let } x = e \text{ in } c, B) \equiv VC(temp := x;\ x := e;\ c;\ x := temp, B)$

$= [x/temp]\ VC(x := e;\ c;\ x := temp, B)$

$= [x/temp][e/x]\ VC(c;\ x := temp, B)$

$= [x/temp][e/x]\ VC(c, [temp/x]B).$

Exercise 4F-3:

Since we know that the bug of the let is that it does not restore the original value of x, we show the following using the buggy let rule:

1. command c is $\text{let } x=1 \text{ in } x := x+1$
2. post-condition B is $\{x=0\}$
3. state σ where $\sigma(x)=0$.
4. $VC(c, B) = [1/x]\ VC(x := x+1, B) = [1/x][2/x]B$. $\sigma \models [1/x][2/x]B$ given $\sigma(x)=0$ and the assignments the variable x.
5. $\langle c, \sigma \rangle \Downarrow \sigma'$ where $\sigma'(x) = 2$.
6. $\sigma' \not\models B$ since $\sigma'(x)=2$ and in B, we have $\{x=0\}$. and $0 \neq 2$.

Exercise 4F-4:

Peer Review ID: 70873991 — enter this when you fill out your peer evaluation via gradescope

2 4F-2 VCGen for Let

- **0 pts** Correct

Peer Review ID: 70873991 — enter this when you fill out your peer evaluation via gradescope

Exercise 4F-2:

That rule for let has a bug where the original value of the variable X is not restored. Since for the let command, this new value e of x has a scope only in the c command.

A correct rule for let needs to restore the original value of x, which we store at a temporary variable named temp.

$VC(\text{let } x=e \text{ in } c, B) \equiv VC(temp := X;\ X := e;\ c;\ X := temp,\ B)$

$= [X/temp]\ VC(X := e;\ c;\ X := temp,\ B)$

$= [X/temp][e/X]\ VC(c;\ X := temp,\ B)$

$= [X/temp][e/X]\ VC(c,\ [temp/X]\ B).$

Exercise 4F-3:

Since we know that the bug of the let is that it does not restore the original value of X, we show the following using the buggy let rule:

1. command c is let $X=1$ in $X := X+1$
2. post-condition B is $\{X=0\}$
3. state σ where $\sigma(X)=0$.
4. $VC(c, B) = [1/X]\ VC(X := X+1, B) = [1/X][2/X]\ B$. $\sigma \models [1/X][2/X]\ B$ given $\sigma(X)=0$ and the assignments the variable x.
5. $\langle c, \sigma \rangle \Downarrow \sigma'$ where $\sigma'(X)=2$.
6. $\sigma' \not\models B$ since $\sigma'(X)=2$ and in B, we have $\{X=0\}$. and $0 \neq 2$.

Exercise 4F-4:

3 4F-3 VCGen Mistakes

- **0 pts** Correct

Peer Review ID: 70873991 — enter this when you fill out your peer evaluation via gradescope

do-while loop is essentially a regular while loop but we perform the command c first. So we have

$$\text{do } c \text{ while } b \equiv c \,;\, \text{while } b \text{ do } c.$$

Therefore, we can build upon the Hoare Logic for while loop as shown in class for do-while:

$$\frac{\vdash \{A\}\ c\ \{B\} \qquad \vdash \{B \land b\}\ c\ \{B\}}{\vdash \{A\}\ \text{do } c \text{ while } b\ \{B \land \neg b\}}$$

Peer Review ID: 70873991 — enter this when you fill out your peer evaluation via gradescope

4 4F-4 Axiomatic Do-While

- **0 pts** Correct

Peer Review ID: 70873991 — enter this when you fill out your peer evaluation via gradescope