1 4F-1 Bookkeeping

- **0 pts** Correct

Exercise 4F-2. VCGen for Let [6 points]. The correct rule for let is:

VC(let x = e in c, B) = VC($y := x$; $x := e$; c; $x := y$,B)

The problem with the original rule was that it has $x = e$ in the post-condition. To rectify this error, I used a sequential series of commands that assigns x to a temporary variable first, then assigns e to x, executes command c, then reassigns x to its original value stored in the temporary variable. Thus, the post condition has x equal to its pre-condition value.

Exercise 4F-3. VCGen Mistakes [6 points]. The following demonstrates the unsoundness of the buggy let rule.

1. $c =$ let $x = 2$ in *skip*
2. B is the post condition of c with $x = 2$
3. σ is a state with $x = 1$
4. $\sigma \models \text{VC}(c, B)$
5. $\langle c, \sigma \rangle \Downarrow \sigma'$
6. $\sigma' \not\models B$ because $x = 1$ in σ' but B requires $x = 2$.

Exercise 4F-4. Axiomatic Do-While [6 points]. The following is a Hoare rule for do c while b.

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

Explanation: "do c while b" is the same as executing c first, then doing "while b do c."

2 4F-2 VCGen for Let

- **0 pts** Correct

Exercise 4F-2. VCGen for Let [6 points]. The correct rule for let is:

VC(let x = e in c, B) = VC($y := x$; $x := e$; c; $x := y$,B)

The problem with the original rule was that it has $x = e$ in the post-condition. To rectify this error, I used a sequential series of commands that assigns x to a temporary variable first, then assigns e to x, executes command c, then reassigns x to its original value stored in the temporary variable. Thus, the post condition has x equal to its pre-condition value.

Exercise 4F-3. VCGen Mistakes [6 points]. The following demonstrates the unsoundness of the buggy let rule.

1. $c =$ let $x = 2$ in *skip*
2. B is the post condition of c with $x = 2$
3. σ is a state with $x = 1$
4. $\sigma \models \text{VC}(c, B)$
5. $\langle c, \sigma \rangle \Downarrow \sigma'$
6. $\sigma' \not\models B$ because $x = 1$ in σ' but B requires $x = 2$.

Exercise 4F-4. Axiomatic Do-While [6 points]. The following is a Hoare rule for do c while b.

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

Explanation: "do c while b" is the same as executing c first, then doing "while b do c."

3 4F-3 VCGen Mistakes

- **0 pts** Correct

Exercise 4F-2. VCGen for Let [6 points]. The correct rule for let is:

VC(let x = e in c, B) = VC($y := x$; $x := e$; c; $x := y$,B)

The problem with the original rule was that it has $x = e$ in the post-condition. To rectify this error, I used a sequential series of commands that assigns x to a temporary variable first, then assigns e to x, executes command c, then reassigns x to its original value stored in the temporary variable. Thus, the post condition has x equal to its pre-condition value.

Exercise 4F-3. VCGen Mistakes [6 points]. The following demonstrates the unsoundness of the buggy let rule.

1. $c =$ let $x = 2$ in *skip*
2. B is the post condition of c with $x = 2$
3. σ is a state with $x = 1$
4. $\sigma \models \text{VC}(c, B)$
5. $\langle c, \sigma \rangle \Downarrow \sigma'$
6. $\sigma' \not\models B$ because $x = 1$ in σ' but B requires $x = 2$.

Exercise 4F-4. Axiomatic Do-While [6 points]. The following is a Hoare rule for do c while b.

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

Explanation: "do c while b" is the same as executing c first, then doing "while b do c."

4 4F-4 Axiomatic Do-While

- **0 pts** Correct