1 4F-1 Bookkeeping

- **0 pts** Correct

Exercise 4F-2. VCGen for Let [6 points]. In class we gave the following rules for the (backward) verification condition generation of assignment and let:

$$\begin{array}{ll} \mathrm{VC}(c_1; c_2, B) & = \mathrm{VC}(c_1, \mathrm{VC}(c_2, B)) \\ \mathrm{VC}(x := e, B) & = [e/x]\ B \\ \mathrm{VC}(\mathsf{let}\ x = e\ \mathsf{in}\ c, B) & = [e/x]\ \mathrm{VC}(c, B) \end{array}$$

That rule for let has a bug. Give a correct rule for let.

The bug in this rule with let is the fact that it doesn't reassign the original value to x after running the command c. The correct rule is shown below:

$$\mathrm{VC}(\mathsf{let}\ x = e\ \mathsf{in}\ c, B) = [x/y][e/x]\ \mathrm{VC}(c, [y/x]B)$$

Exercise 4F-3. VCGen Mistakes [6 points]. Given $\{A\}c\{B\}$ we desire that $A \implies \mathrm{VC}(c, B) \implies \mathrm{WP}(c, B)$. We say that our VC rules are *sound* if $\models \{\mathrm{VC}(c, B)\}\ c\ \{B\}$. Demonstrate the unsoundness of the buggy let rule by giving the following six things:

1. a command c and
2. a post-condition B and
3. a state σ such that
4. $\sigma \models \mathrm{VC}(c, B)$ and
5. $\langle c, \sigma\rangle \Downarrow \sigma'$ but
6. $\sigma' \not\models B$.

We show these following 6 things to show the unsoundness of the let rule.

1. c is let x = 7 in skip
2. B is $x < 8$
3. σ is such that x = 10
4. $\sigma \models \mathrm{VC}(c, B)$ evaluates to $[7/x]$VC(skip, $x < 8$) which evaluates to $7 < 8$ which is true!
5. $\langle c, \sigma\rangle \Downarrow \sigma'$ where σ' has x = 10 since the original value of x is restored after the let command.
6. $\sigma' \models B$ evaluates to $10 < 8$ which is false! So $\sigma' \not\models B$.

2 4F-2 VCGen for Let

- **0 pts** Correct

Exercise 4F-2. VCGen for Let [6 points]. In class we gave the following rules for the (backward) verification condition generation of assignment and let:

$$\begin{array}{ll} \mathrm{VC}(c_1; c_2, B) & = \mathrm{VC}(c_1, \mathrm{VC}(c_2, B)) \\ \mathrm{VC}(x := e, B) & = [e/x]\ B \\ \mathrm{VC}(\mathsf{let}\ x = e\ \mathsf{in}\ c, B) & = [e/x]\ \mathrm{VC}(c, B) \end{array}$$

That rule for let has a bug. Give a correct rule for let.

The bug in this rule with let is the fact that it doesn't reassign the original value to x after running the command c. The correct rule is shown below:

$$\mathrm{VC}(\mathsf{let}\ x = e\ \mathsf{in}\ c, B) = [x/y][e/x]\ \mathrm{VC}(c, [y/x]B)$$

Exercise 4F-3. VCGen Mistakes [6 points]. Given $\{A\}c\{B\}$ we desire that $A \implies \mathrm{VC}(c, B) \implies \mathrm{WP}(c, B)$. We say that our VC rules are *sound* if $\models \{\mathrm{VC}(c, B)\}\ c\ \{B\}$. Demonstrate the unsoundness of the buggy let rule by giving the following six things:

1. a command c and
2. a post-condition B and
3. a state σ such that
4. $\sigma \models \mathrm{VC}(c, B)$ and
5. $\langle c, \sigma\rangle \Downarrow \sigma'$ but
6. $\sigma' \not\models B$.

We show these following 6 things to show the unsoundness of the let rule.

1. c is let x = 7 in skip
2. B is $x < 8$
3. σ is such that x = 10
4. $\sigma \models \mathrm{VC}(c, B)$ evaluates to $[7/x]$VC(skip, $x < 8$) which evaluates to $7 < 8$ which is true!
5. $\langle c, \sigma\rangle \Downarrow \sigma'$ where σ' has x = 10 since the original value of x is restored after the let command.
6. $\sigma' \models B$ evaluates to $10 < 8$ which is false! So $\sigma' \not\models B$.

3 4F-3 VCGen Mistakes

- **0 pts** Correct

Exercise 4F-4. Axiomatic Do-While [6 points]. Write a sound and complete Hoare rule for do c while b. This statement has the standard semantics (e.g., c is executed at least once, before b is tested).

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

Submission. Turn in the formal component of the assignment as a single PDF document via the `gradescope` website. Your name and Michigan email address must appear on the first page of your PDF submission but may not appear anywhere else.

4 4F-4 Axiomatic Do-While

- **0 pts** Correct