1 4F-1 Bookkeeping

- **0 pts** Correct

Peer Review ID: 70803281 — enter this when you fill out your peer evaluation via gradescope

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.

Answer: The new let rule is as follows:

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

This rule replaces the bound variable x in let with e in B. This allows the substitution to be scoped to the let expression only. After the VC has been generated, we must replace e with x to reverse this replacement. This is because e does not have any meaning outside of the let expression and therefore needs to be substituted back.

---

2 4F-2 VCGen for Let

- 0 pts Correct

Peer Review ID: 70803281 — enter this when you fill out your peer evaluation via gradescope

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. $\sigma' \not\models B$.

Answer:

1. $c = \mathsf{let}\ x = 2\ \mathsf{in}\ x := x + 2$
2. $B = \{x \geq 4\}$
3. $\sigma = \{x = 3\}$
4. We can compute $\mathrm{VC}(c, B) = [2/x]\mathrm{VC}(x := x+2, B) = [2/x][x+2/x]B = [2+2/x]B = [4/x]\{x \geq 4\} = \{4 \geq 4\} = \{\mathsf{true}\}$. We know that for any $\sigma \models \mathsf{true}$ always. Therefore, we can conclude that $\sigma \models \mathrm{VC}(c, B)$.
5. We have that $\langle c, \sigma\rangle \Downarrow \sigma'$ where $\sigma' = \sigma = \{x = 3\}$. σ is unaffected by c because x in let only affects x within the let command.
6. We can see that $\sigma' \not\models B$ as $3 \not\geq 4$.

---

3 4F-3 VCGen Mistakes

- 0 pts Correct

Peer Review ID: 70803281 — enter this when you fill out your peer evaluation via gradescope

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).

Answer: This new rule uses the while rule found on slide #23 of lecture 8 (Introduction to Axiomatic Semantics):

$$\frac{\{\mathrm{A}\}\ c\ \{\mathrm{C}\} \quad \{\mathrm{C}\}\ \mathsf{while}\ b\ \mathsf{do}\ c\ \{\mathrm{B}\}}{\{\mathrm{A}\}\ \mathsf{do}\ c\ \mathsf{while}\ b\ \{\mathrm{B}\}}$$

This rule leverages the fact that a do-while loop is identical to a while-do loop, except it executes c once before checking the b condition. We can therefore compose a rule for it (like the one above) by executing c once and then using the same semantics as a while-do loop.

---

4 4F-4 Axiomatic Do-While

- 0 pts Correct

Peer Review ID: 70803281 — enter this when you fill out your peer evaluation via gradescope