1 4F-1 Bookkeeping

- 0 pts Correct

Peer Review ID: 70832359 — 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:

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

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

Solution The command let $x = e$ in c is equivalent to the sequence $x := e; c; x := \sigma(x)$ where σ is the initial state before the sequence. So, using the VC rule for command sequences and assignment, the VC rule for let is

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

2 4F-2 VCGen for Let

- **0 pts** Correct

Peer Review ID: 70832359 — 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 $A \implies \text{VC}(c, B) \implies \text{WP}(c, B)$. We say that our VC rules are *sound* if $\models \{\text{VC}(c, B)\}\ c\ \{B\}$. 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. $\sigma \models \text{VC}(c, B)$ and
5. $\langle c, \sigma\rangle \Downarrow \sigma'$ but
6. $\sigma' \not\models B$.

Solution Let the command c be $\mathsf{let}\ x = 1\ \mathsf{in}\ x := 1$. Let the initial state σ be such that $\sigma(x) = 0$, and let the post-condition be $B = x > 0$. Using the buggy let VC rule, we have that

$$\begin{aligned}
\text{VC}(c, B) &= \text{VC}(\mathsf{let}\ x = 1\ \mathsf{in}\ x := 1, x > 0) \\
&= [1/x]\ \text{VC}(x := 1, x > 0) \\
&= [1/x]\ [1/x]\ x > 0 \\
&= [1/x]\ 1 > 0 \\
&= 1 > 0
\end{aligned}$$

$1 > 0$ is always true, so we have that $\sigma \models 1 > 0 = \text{VC}(c, B)$. However, based on the operational semantics of let, $\langle c, \sigma\rangle \Downarrow \sigma'$ where $\sigma'(x) = \sigma(x) = 0 \not> 0$, so $\sigma' \not\models B$.

3 4F-3 VCGen Mistakes

- **0 pts** Correct

Peer Review ID: 70832359 — 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 $\textsf{do}\ c\ \textsf{while}\ b$. This statement has the standard semantics (e.g., c is executed at least once, before b is tested).

Solution The command $\textsf{do}\ c\ \textsf{while}\ b$ is equivalent to the sequence $c;\ \textsf{while}\ b\ \textsf{do}\ c$. Then, using the Hoare rules for command sequences and while, the Hoare rule for do while is

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

Peer Review ID: 70832359 — enter this when you fill out your peer evaluation via gradescope

4 4F-4 Axiomatic Do-While

- **0 pts** Correct

Peer Review ID: 70832359 — enter this when you fill out your peer evaluation via gradescope