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:

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

Peer Review ID: 316791559 — 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:

1. let $x = 2$ in $x := x + 1$
2. $B : x = 2$
3. $\sigma : x = 0$
4. Applying the buggy let rule:

$$\begin{aligned} \text{VC}(\text{let } x = 2 \text{ in } x := x + 1, x = 2) \\ = [2/x]\text{VC}(x := x + 1, x = 2) \\ = [(x + 1)/x](x = 2) \implies (x + 1) = 2 \\ = [2/x]((x + 1) = 2) \implies 3 = 2 \end{aligned}$$

5. After executing c, x = 3 in σ'.
6. Therefore, $\sigma' \not\models B$, because $x \neq 2$.

Peer Review ID: 316791559 — 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).

Solution: $\{Inv\}$ do c while b $\{Inv \wedge \neg b\}$

Peer Review ID: 316791559 — enter this when you fill out your peer evaluation via gradescope