1 5F-1 Bookkeeping

- 0 pts Correct

Exercise 5F-2. VCGen Do-While [8 points]. We give the backward verification condition formula for the command $\textsf{do}_{\mathit{Inv1},\mathit{Inv2}}\ c\ \textsf{while}\ b$ with respect to a post-condition P. In the below steps, c_{Inv1} corresponds to the command c in state where *Inv1* is true before first executing c. Additionally, $x_1, ..., x_n$ are all of the variables modified in the command c.

$$\begin{aligned}
& \mathrm{VC}(\textsf{do}_{\mathit{Inv1},\mathit{Inv2}}\ c\ \textsf{while}\ b, P) && (1)\\
=\ & \mathrm{VC}(c_{\mathit{Inv1}}\ ;\ \textsf{while}_{\mathit{Inv2}}\ b\ \textsf{do}\ c, P) && (2)\\
=\ & \mathit{Inv1} \wedge \mathrm{VC}(c\ ;\ \textsf{while}_{\mathit{Inv2}}\ b\ \textsf{do}\ c, P) && (3)\\
=\ & \mathit{Inv1} \wedge \mathrm{VC}(c, \mathrm{VC}(\textsf{while}_{\mathit{Inv2}}\ b\ \textsf{do}\ c, P)) && (4)\\
=\ & \mathit{Inv1} \wedge \mathrm{VC}(c, \mathit{Inv2} \wedge (\forall x_1, ..., x_n\ .\ \mathit{Inv2} \implies (b \implies \mathrm{VC}(c, \mathit{Inv2}) \wedge \neg b \implies P))) && (5)
\end{aligned}$$

Step (2) follows from the definition of the command $\textsf{do}\ c\ \textsf{while}\ b$. Specifically, we execute c once, and then the rest of the command is equivalent to $\textsf{while}\ b\ \textsf{do}\ c$. Thus, we can represent $\textsf{do}\ c\ \textsf{while}\ b$ as the sequence $c\ ;\ \textsf{while}\ b\ \textsf{do}\ c$. Step (3) follows from the fact that *Inv1* must be true before executing c the first time. Step (4) follows from the VC rule for sequencing. Finally, step (5) follows from the VC rule for $\textsf{while}\ b\ \textsf{do}\ c$.

2 5F-2 VCGen Do-While

- 0 pts Correct

Peer Review ID: 72925560 — enter this when you fill out your peer evaluation via gradescope

Exercise 5F-3. VCGen Mistakes [20 points]. First, we show that the targaryen rule is incomplete, providing the following:

1. Name of rule: targaryen
2. A: $x \leq 0$
3. B: $x = 6$
4. σ: $[x := 0]$
5. σ': $[x := 6]$
6. c: while $x \leq 5$ do $x := x + 1$
7. $\langle c, \sigma \rangle \Downarrow \sigma'$: We initially have $\sigma = [x := 0]$, and x and is incremented by 1 every iteration of the loop until it is no longer less than or equal to 5. This means that we must have $\sigma' = [x := 6]$ at the end of the loop, meaning that $\langle c, \sigma \rangle \Downarrow \sigma'$.
8. $\sigma \models A$: Because $\sigma = [x := 0]$, it is true that $\sigma \models x \leq 0$.
9. $\sigma' \models B$: Because $\sigma' = [x := 6]$, it is true that $\sigma' \models x = 6$.
10. It is not possible to prove that $\vdash \{A\}\ c\ \{B\}$: Suppose we could prove $\vdash \{A\}\ c\ \{B\}$. Applying the rule of consequence and the targaryen rule (where I is the invariant in the rule), we have

$$\frac{\vdash x \leq 0 \implies I \quad \dfrac{\dfrac{}{\vdash \{I\}\ x := x+1\ \{I\}}}{\vdash \{I\}\ \text{while } x \leq 5 \text{ do } x := x+1\ \{I \land x > 5\}} \quad \vdash I \land x > 5 \implies x = 6}{\vdash \{x \leq 0\}\ \text{while } x \leq 5 \text{ do } x := x+1\ \{x = 6\}}.$$

To prove that $I \land x > 5 \implies x = 6$, we must have $I = (x \leq 6)$. Otherwise, we would have insufficient evidence to conclude that $x = 6$ while also being able to infer $x \leq 0 \implies I$. Given that $I = (x \leq 6)$, we cannot prove $\{I\}\ x := x+1\ \{I\}$ (at the top of the derivation tree). This is because if $x = 6$ initially, then it must be that $x > 6$ after executing $x := x + 1$. However, this violates the post-condition $x \leq 6$. Thus, we cannot prove that $\vdash \{A\}\ c\ \{B\}$, meaning that the targaryen rule is incomplete.

Next, we show that the stark rule is incomplete, providing the following (items 1-9 are the same as in the above demonstration for the targaryen rule):

1. Name of rule: stark
2. A: $x \leq 0$
3. B: $x = 6$

4. σ: $[x := 0]$

5. σ': $[x := 6]$

6. c: $\textsf{while } x \leq 5 \textsf{ do } x := x + 1$

7. $\langle c, \sigma \rangle \Downarrow \sigma'$: We initially have $\sigma = [x := 0]$, and x and is incremented by 1 every iteration of the loop until it is no longer less than or equal to 5. This means that we must have $\sigma' = [x := 6]$ at the end of the loop, meaning that $\langle c, \sigma \rangle \Downarrow \sigma'$.

8. $\sigma \models A$: Because $\sigma = [x := 0]$, it is true that $\sigma \models x \leq 0$.

9. $\sigma' \models B$: Because $\sigma' = [x := 6]$, it is true that $\sigma' \models x = 6$.

10. It is not possible to prove that $\vdash \{A\}\ c\ \{B\}$: Suppose we could prove $\vdash \{A\}\ c\ \{B\}$. Applying the rule of consequence and the stark rule (where I is the invariant in the rule), we have

$$\dfrac{\vdash x \leq 0 \implies I \quad \dfrac{\overline{\vdash \{I \wedge x \leq 5\}\ x := x + 1\ \{I\}}}{\vdash \{I\}\ \textsf{while } x \leq 5 \textsf{ do } x := x + 1\ \{I\}} \quad \vdash I \implies x = 6}{\vdash \{x \leq 0\}\ \textsf{while } x \leq 5 \textsf{ do } x := x + 1\ \{x = 6\}}.$$

We could again let $I = (x \leq 6)$. In fact, we could choose any integer y greater than or equal to 6 and let $I = (x \leq y)$, as these will all be valid invariants for the loop. None of these invariants, however, will be enough to make the conclusion that $x = 6$ in the application of the rule of consequence. If the post-condition $\{I\}$ of the upper while command included the negation $x > 5$ of the loop guard, we could let $I = (x \leq 6)$ and conclude from $x \leq 6 \wedge x > 5$ that $x = 6$. Because $x > 5$ is missing from the post-condition of the while command, there is no way to conclude that $x = 6$. In other words, there is no assignment of I that satisfies both $x \leq 0 \implies I$ and $I \implies x = 6$; we need to use the fact that $x > 5$ to conclude that $x = 6$. Because of this, we cannot prove that $\vdash \{A\}\ c\ \{B\}$, meaning that the stark rule is incomplete.

3 5F-3 VCGen Mistakes

- 0 pts Correct