1 5F-1 Bookkeeping

- **- 0 pts** Correct

Peer Review ID: 72966034 — enter this when you fill out your peer evaluation via gradescope

5F-2. VCGen Do-While

The formula for $\texttt{do}_{Inv}$ is very reminiscent of the normal $\texttt{while}_{Inv}$ formula, with a key change. Because the `do` form evaluates the command once before ever checking the loop condition, and Inv "must be true before each evaluation of the predicate b", we need to ensure that Inv holds after the first execution of the command. We do this by invoking VC on c, and conjunct Inv with the loop formula in the postcondition:

$$VC(\texttt{do}_{Inv}\ c\ \texttt{while}\ b, P) = VC(c, Inv \wedge (\forall x_1 ... x_n.\ Inv \implies (b \implies VC(c, Inv) \wedge \neg b \implies P))$$

5F-3. VCGen Mistakes

1. We'll show the incompleteness of the "targaryen" rule.
2. $A = x < 3$
3. $B = x < 3 \wedge true$
4. σ such that $\sigma(x) = 2$
5. σ' such that $\sigma'(x) = 2$
6. $c = \texttt{while}\ false\ \texttt{do}\ x := 3$
7. Within the form $\langle c, \sigma \rangle \Downarrow \sigma'$, $\sigma' = \sigma$:

$$\dfrac{\dfrac{}{\langle false, \sigma \rangle \Downarrow false}}{\langle \texttt{while}\ false\ \texttt{do}\ c, \sigma \rangle \Downarrow \sigma}$$

8. $\sigma(x) = 2 \wedge 2 < 3 \implies \sigma(x) < 3$. Thus, $\sigma \models x < 3$, and trivially $\sigma \models x < 3 \wedge true$.
9. $\sigma' = \sigma \implies \sigma'(x) = \sigma(x)$. By the same logic as in 8., $\sigma' \models x < 3 \wedge true$.
10. However, using the "targaryen" rule, we fail to prove $\vdash \{A\}\ c\ \{B\}$:

$$\dfrac{\dfrac{\vdash \{x < 3 \implies 3 < 3\} \leftarrow \texttt{FALSE!} \qquad \dfrac{}{\vdash \{3 < 3\}\ x := 3\ \{x < 3\}}}{\vdash \{x < 3\}\ x := 3\ \{x < 3\}}}{\vdash \{x < 3\}\ \texttt{while}\ false\ \texttt{do}\ x := 3\ \{x < 3\}}$$

As labeled, attempting to derive the rule requires that we make a false statement - $x < 3$ does not imply the clearly false $3 < 3$. On a higher level, we can clearly see that the condition $x < 3$ cannot hold across the command $x := 3$. And yet, in the case that we do zero iterations, the condition holds across the loop.

---

1. We'll show the incompleteness of the "lannister" rule.
2. $A = x = 5$
3. $B = y = 5$
4. σ such that $\sigma(x) = 5 \wedge \sigma(y) = 5$
5. σ' such that $\sigma(x) = 5 \wedge \sigma(y) = 5$
6. $c = \texttt{while}\ x \neq y\ \texttt{do}\ x := 4$

2 5F-2 VCGen Do-While

- 0 pts Correct

Peer Review ID: 72966034 — enter this when you fill out your peer evaluation via gradescope

5F-2. VCGen Do-While

The formula for $\texttt{do}_{Inv}$ is very reminiscent of the normal $\texttt{while}_{Inv}$ formula, with a key change. Because the $\texttt{do}$ form evaluates the command once before ever checking the loop condition, and Inv "must be true before each evaluation of the predicate b", we need to ensure that Inv holds after the first execution of the command. We do this by invoking VC on c, and conjunct Inv with the loop formula in the postcondition:

$$VC(\texttt{do}_{Inv}\ c\ \texttt{while}\ b, P) = VC(c, Inv \land (\forall x_1 ... x_n.\ Inv \implies (b \implies VC(c, Inv) \land \neg b \implies P))$$

5F-3. VCGen Mistakes

1. We'll show the incompleteness of the "targaryen" rule.
2. $A = x < 3$
3. $B = x < 3 \land true$
4. σ such that $\sigma(x) = 2$
5. σ' such that $\sigma'(x) = 2$
6. $c = \texttt{while}\ false\ \texttt{do}\ x := 3$
7. Within the form $\langle c, \sigma \rangle \Downarrow \sigma'$, $\sigma' = \sigma$:

$$\frac{\overline{\langle false, \sigma \rangle \Downarrow false}}{\langle \texttt{while}\ false\ \texttt{do}\ c, \sigma \rangle \Downarrow \sigma}$$

8. $\sigma(x) = 2 \land 2 < 3 \implies \sigma(x) < 3$. Thus, $\sigma \models x < 3$, and trivially $\sigma \models x < 3 \land true$.
9. $\sigma' = \sigma \implies \sigma'(x) = \sigma(x)$. By the same logic as in 8., $\sigma' \models x < 3 \land true$.
10. However, using the "targaryen" rule, we fail to prove $\vdash \{A\}\ c\ \{B\}$:

$$\frac{\dfrac{\vdash \{x < 3 \implies 3 < 3\} \leftarrow \texttt{FALSE!} \qquad \overline{\vdash \{3 < 3\}\ x := 3\ \{x < 3\}}}{\vdash \{x < 3\}\ x := 3\ \{x < 3\}}}{\vdash \{x < 3\}\ \texttt{while}\ false\ \texttt{do}\ x := 3\ \{x < 3\}}$$

As labeled, attempting to derive the rule requires that we make a false statement - $x < 3$ does not imply the clearly false $3 < 3$. On a higher level, we can clearly see that the condition $x < 3$ cannot hold across the command $x := 3$. And yet, in the case that we do zero iterations, the condition holds across the loop.

---

1. We'll show the incompleteness of the "lannister" rule.
2. $A = x = 5$
3. $B = y = 5$
4. σ such that $\sigma(x) = 5 \land \sigma(y) = 5$
5. σ' such that $\sigma(x) = 5 \land \sigma(y) = 5$
6. $c = \texttt{while}\ x \neq y\ \texttt{do}\ x := 4$

7. Within the form $\langle c, \sigma\rangle \Downarrow \sigma'$, $\sigma' = \sigma$:

$$\cfrac{\cfrac{\overline{\langle x, \sigma\rangle \Downarrow 5} \qquad \overline{\langle y, \sigma\rangle \Downarrow 5}}{\langle x \neq y, \sigma\rangle \Downarrow \mathit{false}}}{\langle \texttt{while}\ x \neq y\ \texttt{do}\ x := x - 1, \sigma\rangle \Downarrow \sigma}$$

8. We defined $\sigma(x) = 5$, so $\sigma \models x = 5$.

9. Since $\sigma' = \sigma$, and we defined $\sigma(y) = 5$, $\sigma \models y = 5$.

10. However, using the "lannister" rule, we fail to prove $\vdash \{A\}\ c\ \{B\}$. My derivation tree got way too big, so I'll explain it verbally:

    - We're trying to show $\vdash \{x = 5\}\ x := 4\ \{x \neq y \implies x = 5 \wedge x = y \implies y = 5\}$.
    - We assume the precondition to be true (as it is before any iterations), so $x = 5$.
    - Evaluating the command, $x := 4$, binds x to 4, however.
    - y is never modified, and remains equal to 5, so $x \neq y$.
    - $x \neq y \implies x = 5$. But, we just bound x to 4, so we have a true statement ($x \neq y$) implying a false statement ($x = 5$), so this is false.
    - By the definition of conjunction, this means the postcondition must be false.
    - So, the implicant of the rule is false, and we fail to prove the implicand.
    - But, we found above that the postcondition *does* hold when we evaluate this command. A starting state exists such that the implicand is ultimately true, even though we could not prove it.

    Thus, the "lannister" rule is incomplete.

3 5F-3 VCGen Mistakes

- 0 pts Correct

Peer Review ID: 72966034 — enter this when you fill out your peer evaluation via gradescope