1 5F-1 Bookkeeping

- **0 pts** Correct

Exercise 5F-2. VCGen Do-While [8 points]. Choose exactly *one* of the two options below. (If you are not certain, pick the first. The answers end up being equivalent, but the first may be easier to grasp for some students and the second easier to grasp for others.)

- Give the (backward) verification condition formula for the command $\mathsf{do}_{Inv}\ c\ \mathsf{while}\ b$ with respect to a post-condition P. The invariant Inv is true before each evaluation of the predicate b. Your answer may not be defined in terms of VC(while...).
- Give the (backward) verification condition formula for the command $\mathsf{do}_{Inv1,Inv2}\ c\ \mathsf{while}\ b$ with respect to a post-condition P. The invariant *Inv1* is true before c is first executed. The invariant *Inv2* is true before each evaluation of the loop predicate b. Your answer may not be defined in terms of VC(while...).

Answer 5F-2

I chose the first option.

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

Because c will be executed at least once, we know that b will be evaluated at least once. Because Inv needs to be true before b is evaluated, we must have that $VC(c, Inv)$ is true on entry.

On every iteration, Inv must hold, which means that if b evaluates to true, then we are going back to the top of the loop, where $VC(c, Inv)$ must be true. Otherwise, the post-condition P must hold because we are exiting the loop.

---

2 5F-2 VCGen Do-While

- **0 pts** Correct

Exercise 5F-3. VCGen Mistakes [20 points]. Consider the following three alternate while Hoare rules (named lannister, stark, and targaryen):

$$\frac{\vdash \{X\}\ c\ \{b \implies X\ \wedge\ \neg b \implies Y\}}{\vdash \{b \implies X\ \wedge\ \neg b \implies Y\}\ \mathsf{while}\ b\ \mathsf{do}\ c\ \{Y\}}\ \textsf{lannister} \qquad \frac{\vdash \{X\ \wedge\ b\}\ c\ \{X\}}{\vdash \{X\}\ \mathsf{while}\ b\ \mathsf{do}\ c\ \{X\}}\ \textsf{stark}$$

$$\frac{\vdash \{X\}\ c\ \{X\}}{\vdash \{X\}\ \mathsf{while}\ b\ \mathsf{do}\ c\ \{X\ \wedge\ \neg b\}}\ \textsf{targaryen}$$

All three rules are sound but incomplete. Choose **two** incomplete rules. For each chosen rule provide the following:

1. the name of the rule and
2. A and
3. B and
4. σ and
5. σ' and
6. c such that
7. $\langle c, \sigma\rangle \Downarrow \sigma'$ and
8. $\sigma \models A$ and
9. $\sigma' \models B$ but
10. it is not possible to prove $\vdash \{A\}\ c\ \{B\}$.

Flavor text: Incompleteness in an axiomatic semantics or type system is typically not as dire as unsoundness. An incomplete system cannot prove all possible properties or handle all possible programs. Many research results that claim to work for the C language, for example, are actually incomplete because they do not address `setjmp`/`longjmp` or bitfields. (Many of them are also unsound because they do not correctly model unsafe casts, pointer arithmetic, or integer overflow.)

Answer 5F-3

Targaryen
If the Targaryen rule was complete, then the following would be valid, but we can see that the premise is not provable.

$$\frac{\vdash \{x = 1\}\ x := 0\ \{x = 1\}}{\vdash \{x = 1\}\ \mathsf{while}\ \mathit{false}\ \mathsf{do}\ x := 0\ \{x = 1\ \wedge\ \neg \mathit{false}\}}$$

2. A is $x = 1$
3. B is $x = 1 \ \wedge \neg$ false
4. $\sigma(x) = 1$
5. $\sigma'(x) = 1$
6. c is (while false do x := 0)
7. $\langle$ while false do x := 0, $\sigma\rangle \Downarrow \sigma'$ is a true statement because $\sigma(x) = 1$ and the command doesn't modify x, which means that $\sigma'(x) = 1$.
8. $\sigma \models$ A because $\sigma(x) = 1$
9. $\sigma' \models$ B because $\sigma'(x) = 1$
10. However, it is not possible to prove $\vdash \{x = 1\}\ x := 0\ \{x = 1\}$ because after x is assigned to 0, the resulting state cannot have x assigned to 1.

Lannister
If the Lannister rule was complete, then the following would be valid, but we can see that the premise is not provable.

$$\frac{\vdash \{x = 2\}\ x := 0\ \{false \implies x = 2 \ \wedge \ \neg false \implies x = 1\}}{\vdash \{false \implies x = 2 \ \wedge \ \neg false \implies x = 1\}\ \mathsf{while}\ false\ \mathsf{do}\ x := 0\ \{x = 1\}}$$

2. A is $false \implies x = 2 \ \wedge \ \neg false \implies x = 1$
3. B is x = 1
4. $\sigma(x) = 1$
5. $\sigma'(x) = 1$
6. c is (while false do x := 0)
7. $\langle$ while false do x := 0, $\sigma\rangle \Downarrow \sigma'$ is a true statement because $\sigma(x) = 1$ and the command doesn't modify x, which means $\sigma'(x) = 1$.
8. $\sigma \models$ A because $\sigma(x) = 1$
9. $\sigma' \models$ B because $\sigma'(x) = 1$
10. However, it is not possible to prove $\vdash \{x = 2\}\ x := 0\ \{false \implies x = 2 \ \wedge \ \neg false \implies x = 1\}$ because after x is assigned to 0, the resulting state cannot have x assigned to 1.

Submission. Turn in the formal component of the assignment as a single PDF document via the `gradescope` website. Your name and Michigan email address must appear on the first page of your PDF submission but may not appear anywhere else.

3 5F-3 VCGen Mistakes

- **0 pts** Correct