1 5F-1 Bookkeeping

- **0 pts** Correct

Peer Review ID: 72939797 — enter this when you fill out your peer evaluation via gradescope

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 $\mathsf{do}_{\mathit{Inv}}\ c\ \mathsf{while}\ b$ 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 $\mathsf{do}_{\mathit{Inv1},\mathit{Inv2}}\ c\ \mathsf{while}\ b$ 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 $VC(c, Inv) \wedge (\forall x_1 ... x_n . Inv => (e => VC(c, Inv)) \wedge (\neg e => B))$ This is similar to the while command, but includes a VC for the command with the invariant before evaluations of e.

2 5F-2 VCGen Do-While

- **0 pts** Correct

Peer Review ID: 72939797 — enter this when you fill out your peer evaluation via gradescope

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

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

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

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 I choose Stark and Targaryen.

1. Stark
2. $A ==$ true
3. $B == \{x >= 10\}$
4. $\sigma == \{\mathrm{x} := 2\}$

5. $\sigma' == \{\text{x} := 10\}$

6. $c ==$ while $x < 10$ do x := x + 1

7. $\langle c, \sigma \rangle \Downarrow \sigma'$

8. $\sigma \models A$ (true) and

9. $\sigma' \models B$ (true) but

10. it is not possible to prove $\vdash \{A\}\ c\ \{B\}$, as this rule does not assert that the postcondition of while b do c must have ¬ b, unlike the standard while Hoare rule. In this case, X is ”true”, so the immediate postcondition of the while rule as given is true. For this to have B as a postcondition, this would require that we be able to prove that true implies $x >= 10$ with the rule of consequence, which does not work.

1. Targaryen

2. $A == \{\text{x} := 11\}$

3. $B == \{x >= 10\}$

4. $\sigma == \{\text{x} := 11\}$

5. $\sigma' == \{\text{x} := 11\}$

6. $c ==$ while $x < 10$ do x := x + 1

7. $\langle c, \sigma \rangle \Downarrow \sigma'$

8. $\sigma \models A$ and

9. $\sigma' \models B$ but

10. it is not possible to prove $\vdash \{A\}\ c\ \{B\}$. X is ”true”, and this rule does not assert that the precondition of c in while b do c must have b. That mean that when we try to derive it, we get {x := 11 } x := x + 1 {x := 11}; we cannot make this derivation, so we cannot prove the overall case

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

Peer Review ID: 72939797 — enter this when you fill out your peer evaluation via gradescope