1 5F-1 Bookkeeping

- 0 pts Correct

Peer Review ID: 72950584 — enter this when you fill out your peer evaluation via gradescope

All subsequent answers should appear after the first page of your submission and may be shared publicly during peer review.

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...).

I'll pick the first option.

$$\begin{aligned} & VC(\mathsf{do}_{\mathit{Inv}}\ c\ \mathsf{while}\ b, P) \\ =& VC(c, VC(\mathsf{while}_{\mathit{Inv}}\ b\ \mathsf{do}\ c, P)) \\ =& VC(c, \mathit{Inv} \land (\forall x_1 \ldots x_n . \mathit{Inv} \implies (b \implies VC(c, \mathit{Inv}) \land \neg b \implies P))) \end{aligned} \tag{1}$$

Peer Review ID: 72950584 — enter this when you fill out your peer evaluation via gradescope

2 5F-2 VCGen Do-While

- 0 pts Correct

Peer Review ID: 72950584 — 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):

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

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

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.)

For the first incomplete rule:

1. lannister
2. let A be $x > 0 \implies \mathsf{true} \wedge x \leq 0 \implies x = 0$
3. let B be $x = 0$
4. let σ be $x = 1$
5. let σ' be $x = 0$
6. let c be $\mathsf{while}\ x > 0\ \mathsf{do}\ x := x - 1$
7. $\langle c, \sigma \rangle \Downarrow \sigma'$ because

   $$\frac{\langle x > 0, \sigma \rangle \Downarrow \mathsf{true} \quad \langle x := x - 1, \sigma \rangle \Downarrow \sigma' \quad \dfrac{\langle x > 0, \sigma' \rangle \Downarrow \mathsf{false}}{\langle\ \mathsf{while}\ x > 0\ \mathsf{do}\ x := x - 1, \sigma' \rangle \Downarrow \sigma'}}{\langle \mathsf{while}\ x > 0\ \mathsf{do}\ x := x - 1, \sigma \rangle \Downarrow \sigma'}$$

   where $\sigma' = \sigma[\sigma(x) := 0]$
8. $\sigma \models A$ because $\sigma[\sigma(x) = 1] \models x > 0 \implies x = 1 \wedge x \leq 0 \implies x = 0$ (note that false imples anything for the second half of A)
9. $\sigma' \models B$ because $\sigma[\sigma(x) = 0] \models x = 0$
10. it is not possible to prove $\vdash \{A\}\ c\ \{B\}$ because we would stuck at

    $$\frac{\dfrac{}{\vdash \{\mathsf{true}\}\ x := x - 1\ \{x > 0 \implies \mathsf{true}\ \wedge\ x \leq 0 \implies x = 0\}}}{\vdash \{x > 0 \implies \mathsf{true}\ \wedge\ x \leq 0 \implies x = 0\}\ \mathsf{while}\ x > 0\ \mathsf{do}\ x := x - 1\ \{x = 0\}}$$

    Since true implies nothing about x, we cannot verify $x \leq 0 \implies x = 0$.

For the second incomplete rule:

1. $\mathsf{targaryen}$
2. let A be true
3. let B be $\mathsf{true} \wedge x \leq 1$
4. let σ be $\sigma(x) = 1$
5. let $\sigma' = \sigma$
6. let c be $\mathsf{while}\ x > 1\ \mathsf{do}\ x := x - 1$
7. $\langle c, \sigma \rangle \Downarrow \sigma'$ because

$$\frac{\langle x > 1, \sigma \rangle \Downarrow \mathsf{false}}{\langle \mathsf{while}\ x > 1\ \mathsf{do}\ x := x - 1, \sigma \rangle \Downarrow \sigma'} \text{ where } \sigma' = \sigma$$

8. $\sigma \models A$ because $\sigma \models \mathsf{true}$
9. $\sigma' \models B$ because $\sigma'[\sigma'(x) = 1] \models \mathsf{true} \wedge x \leq 1$
10. it is not possible to prove $\vdash \{A\}\ c\ \{B\}$ because according to the rule,

$$\frac{\vdash \{\mathsf{true}\}\ x := x - 1\ \{\mathsf{true}\}}{\vdash \{\mathsf{true}\}\ \mathsf{while}\ x > 1\ \mathsf{do}\ x := x - 1\ \{\mathsf{true} \wedge x \leq 1\}}$$

true does not imply anything about x, and without a precondition about x, we cannot know if $x \leq 1$ holds with $x := x - 1$ only. Therefore, there is no way we can get any information about x from the our hypothesis, and the post-condition $\mathsf{true} \wedge \mathsf{x} \leq 1$ cannot be verified.

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: 72950584 — enter this when you fill out your peer evaluation via gradescope