2 Exercise 5F-2. VCGen Do-While

I will give a VC formula for $\text{do}_{Inv}\ c$ while b. Inv is true before each evaluation of b.

Any do-while loop can be converted into a while loop where the statement is run once prior to beginning the loop body. Thus, this statement can be considered as $c;$ while $_{Inv}b$ do c.

We can break our generated precondition into various properties that must hold. For the invariant to hold on entry, we need the precondition $\text{VC}(c, Inv)$. The invariant must hold between iterations, so $Inv \implies (b \implies \text{VC}(c, Inv))$ is also required. Upon exit, the post condition needs to hold, so $\neg b \implies P$.

All together, our rule is then:

$$\text{VC}(\text{do}_{Inv}\ c \text{ while } b) = \text{VC}(c, Inv) \wedge Inv \implies (b \implies \text{VC}(c, Inv)) \wedge \neg b \implies P$$

3 Exercise 5F-3.VCGen Mistakes

I will show incompleteness for the stark and targaryen rules.

3.1 Stark

$$\frac{\vdash \{X \wedge b\} c \{X\}}{\vdash \{X\} \text{ while } b \text{ do } c \{X\}}$$

Consider the following conditions:

$$\begin{aligned}
&A := y = 5 \\
&B := A \\
&c := \text{while } \ \text{k} = 0 \ \ \text{do } \ \text{y} = 10; \\
&\sigma \text{ such that } \sigma(y) := 5, \sigma(k) := 1 \\
&\sigma' \text{ such that } \sigma' := \sigma
\end{aligned}$$

Because $k \neq 0$, the state is not modified so $\langle c, \sigma \rangle \downarrow \sigma'$. Thus,B holds on σ'.

$\{A \wedge k = 0\} y = 10; \{B\}$ does not hold, thus this correct statement cannot be proven, so the Stark rule is incomplete.

3.2 Targaryen

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

Question assigned to the following page: 3

Consider the following conditions:

$$A := x = 5 \wedge k = 1$$
$$B := x = 5$$
$$c := \text{ while } b = 1 \text{ do } \text{ if } k = 1 \text{ then } b := 0; \text{ else } x = 10;$$
$$\sigma \text{ such that } \sigma(x) = 5, \sigma(k) = 1, \sigma(b) = 1$$
$$\sigma' \text{ such that } \sigma'(x) = 5, \sigma'(k) = 1, \sigma'(b) = 0$$

The loop iterates for 1 cycle, taking the branch where b is set to 0, and leaving the other values unmodified. Thus, σ, σ' accurately represent the initial and ending states of the execution. A holds is σ, and B holds in σ'.

However, because of the branch in the loop body, it is not possible to prove this statement true using the Targaryen rule. Thus, the Targaryen rule is incomplete.