1 5F-1 Bookkeeping

- 0 pts Correct

2 Exercise 5F-2. VCGen Do-While [8 points].

First, we consider the command $do_{Inv1}\ \ c\ \ while\ \ b$. This can be broken down into three parts: $assert(Inv1)\,;c\,;while_{Inv2}\ \ b\ \ do\ \ c$

We introduce a new loop invariant for the while loop, Inv2. This will allow us to check Inv1 before and after c is executed. Additionally, we can present the first execution of the command c. Originally, Inv1 must be true for each iteration of the while loop. Therefore, Inv2 will imply Inv1. $(Inv1 \wedge Inv2 \Rightarrow Inv1)$

We use $x_1, ..., x_n$ to represent the variables modified in c. The result is:

$$VC(assert(Inv1 \wedge Inv2 \Rightarrow Inv1),\ \ VC(c\,;while_{Inv2}\ \ b\ \ do\ \ c,\ B))$$
$$Inv1 \wedge Inv2 \Rightarrow Inv1\ \ \wedge\ \ VC(c\,,VC(while_{Inv2}\ \ b\ \ do\ \ c,\ B))$$
$$Inv1 \wedge Inv2 \Rightarrow Inv1\ \ \wedge\ \ VC(c\,,Inv2 \wedge (\forall x_1, x_2, ..., x_n\ Inv2 \Rightarrow (b \Rightarrow VC(c, Inv2)) \wedge\ \neg b \Rightarrow B))$$

3 Exercise 5F-3. VCGen Mistakes [20 points].

3.1 Stark Rule

The problem of stark rule is that it does not assume $\neg b$ even after the loop terminates.

1. stark
2. A: $x < 3$
3. B: x=6
4. $\sigma(x) = 0$
5. $\sigma'(x) = 6$
6. c: while x < 6 do x := x+1
7. $< c, \sigma > \Downarrow \sigma'$
8. $\sigma \models A$
9. $\sigma' \models B$
10. It is impossible to prove $\{A\}$ while x < 6 do x := x+1 $\{B\}$ using stark rule

Here, we will prove 10 by contradiction. If it is possible, we have a derivation D:

$$D :: \vdash \{x < 3\}\ while\ x\ \ < 6\ do\ x := x + 1\ \{x = 6\}$$

By inversion, the last rule used in D can be either stark rule or the rule of consequence.

If the last rule is stark, we notice that the pre-condition and post-condition are not the same. Therefore it contradicts the definition of stark rule.

2 5F-2 VCGen Do-While

- 0 pts Correct

Peer Review ID: 72958911 — enter this when you fill out your peer evaluation via gradescope

2 Exercise 5F-2. VCGen Do-While [8 points].

First, we consider the command $do_{Inv1}\ \ c\ \ while\ \ b$. This can be broken down into three parts: $assert(Inv1)\,;c\,;while_{Inv2}\ \ b\ \ do\ \ c$

We introduce a new loop invariant for the while loop, Inv2. This will allow us to check Inv1 before and after c is executed. Additionally, we can present the first execution of the command c. Originally, Inv1 must be true for each iteration of the while loop. Therefore, Inv2 will imply Inv1. $(Inv1 \wedge Inv2 \Rightarrow Inv1)$

We use $x_1, ..., x_n$ to represent the variables modified in c. The result is:

$$VC(assert(Inv1 \wedge Inv2 \Rightarrow Inv1),\ \ VC(c\,;while_{Inv2}\ \ b\ \ do\ \ c,\ B))$$
$$Inv1 \wedge Inv2 \Rightarrow Inv1\ \ \wedge\ \ VC(c\,,VC(while_{Inv2}\ \ b\ \ do\ \ c,\ B))$$
$$Inv1 \wedge Inv2 \Rightarrow Inv1\ \ \wedge\ \ VC(c\,,Inv2 \wedge (\forall x_1, x_2, ..., x_n\ Inv2 \Rightarrow (b \Rightarrow VC(c, Inv2)) \wedge\ \neg b \Rightarrow B))$$

3 Exercise 5F-3. VCGen Mistakes [20 points].

3.1 Stark Rule

The problem of stark rule is that it does not assume $\neg b$ even after the loop terminates.

1. stark
2. A: $x < 3$
3. B: x=6
4. $\sigma(x) = 0$
5. $\sigma'(x) = 6$
6. c: while x < 6 do x := x+1
7. $< c, \sigma > \Downarrow \sigma'$
8. $\sigma \models A$
9. $\sigma' \models B$
10. It is impossible to prove $\{A\}$ while x < 6 do x := x+1 $\{B\}$ using stark rule

Here, we will prove 10 by contradiction. If it is possible, we have a derivation D:

$$D :: \vdash \{x < 3\}\ while\ x\ \ < 6\ do\ x := x + 1\ \{x = 6\}$$

By inversion, the last rule used in D can be either stark rule or the rule of consequence.

If the last rule is stark, we notice that the pre-condition and post-condition are not the same. Therefore it contradicts the definition of stark rule.

If the last rule is rule of consequence, D should be

$$\frac{D_1 :: \vdash \{x < 3\} \Rightarrow P \quad \dfrac{D_4 :: \vdash \{P \wedge x < 6\}\ x := x+1\ \{P\}}{D_2 :: \vdash \{P\}\ while\ x\ < 6\ do\ x := x+1\ \{P\}} \quad D_3 :: \ \vdash P \Rightarrow \{x = 6\}}{\vdash \{x < 3\}\ while\ x\ < 6\ do\ x := x+1\ \{x = 6\}}$$

It is impossible to find such P that $\vdash \{x < 3\} \Rightarrow P \Rightarrow \{x = 6\}$.

We have shown the contradiction in both two cases. Therefore, it is impossible to prove $\{A\}$ while x < 6 do x := x+1 $\{B\}$ using stark rule.

3.2 Targaryen Rule

The problem of targaryen rule is that it does not allow you to assume assume b inside the loop.

1. targaryen
2. A: $x \leq 6$
3. B: x=6
4. $\sigma(x) = 0$
5. $\sigma'(x) = 6$
6. c: while x < 6 do x := x+1
7. $< c, \sigma > \Downarrow \sigma'$
8. $\sigma \models A$
9. $\sigma' \models B$
10. It is impossible to prove $\{A\}$ while x $\leq$ 6 do x := x+1 $\{B\}$ using targaryen rule

Here, we will prove 10 by contradiction. If it is possible, we have a derivation D:

$$D :: \vdash \{x \leq 6\}\ while\ x\ < 6\ do\ x := x+1\ \{x = 6\}$$

By inversion, the last rule used in D can be either targaryen rule or the rule of consequence.

If the last rule is targaryen rule, we notice that the post-condition is not textually equal to the precondition $\wedge \neg b$. Therefore it contradicts the definition of targaryen rule.

If the last rule is rule of consequence, D should be

$$\frac{D_1 :: \vdash \{x \leq 6\} \Rightarrow P \quad \dfrac{D_4 :: \vdash \{P\}\ x := x+1\ \{P\}}{D_2 :: \vdash \{P\}\ while\ x\ < 6\ do\ x := x+1\ \{P \wedge \neg x < 6\}} \quad D_3 :: \ \vdash \{P \wedge \neg x < 6\} \Rightarrow \{x = 6\}}{\vdash \{x \leq 6\}\ while\ x\ < 6\ do\ x := x+1\ \{x = 6\}}$$

We can observe that P has to be $x \leq 6$ given that $\vdash \{x \leq 6\} \Rightarrow P$ and $\{P \wedge \neg x < 6\} \Rightarrow \{x = 6\}$. However, D4 cannot exist by soundness for the chosen P when x=6.

We have shown the contradiction in both two cases. Therefore, it is impossible to prove $\{A\}$ while x < 6 do x := x+1 $\{B\}$ using targaryen rule.

3 5F-3 VCGen Mistakes

- 0 pts Correct