1 4F-1 Bookkeeping

- 0 pts Correct

Exercise 4F-2

From previous homework, we know that the command `let` $x = e$ `in` c is equivalent to the sequence of commands $x' := x; x := e; c; x := x'$ where x' is a fresh variable. So, we can compute the verification condition of `let` in terms of the verification conditions of these other commands:

$$\begin{aligned} VC(\texttt{let } x = e \texttt{ in } c, B) &= VC(x' := x; x := e; c; x := x', B) \\ &= VC(x' := x, VC(x := e, VC(c, VC(x := x', B)))) \\ &= [x/x'][e/x]VC(c, [x'/x]B). \end{aligned}$$

Exercise 4F-3

Let c be the command `let` $x = 2$ `in skip`, let B be the condition $x = 2$, let σ be the state $(x \mapsto 1)$. Then, based on the buggy rule for let, we have

$$\begin{aligned} VC(c, B) &= VC(\texttt{let } x = 2 \texttt{ in skip}, \text{“}x = 2\text{”}) \\ &= [2/x]VC(\texttt{skip}, \text{“}x = 2\text{”}) \\ &= [2/x]\text{“}x = 2\text{”} \\ &= \text{“}2 = 2\text{”}. \end{aligned}$$

Clearly $\sigma \vDash VC(c, B)$, because “$2 = 2$” is true regardless of the value of any variable. However, based on our operational semantics rules for `let`, we have $\langle c, \sigma \rangle \Downarrow \sigma$. That is, `let` $x = 2$ `in skip` doesn't change σ because `skip` doesn't change σ.

Finally we have $\sigma \nvDash$ “$x = 2$” because, as defined, $\sigma(x) = 1$.

Exercise 4F-4

We claim that `do` c `while` b is equivalent to executing c once and then running c in a normal `while` loop (i.e. `do` c `while` $b \sim c$; `while` b `do` c). To see this, it suffices to show that they both execute c the same number of times. But this is true because both have the behavior that, after one execution of c they check b and continue looping if and only if b was true. Using this, we can easily write a Hoare rule for `do-while` in terms of the Hoare rules we already know for sequencing and `while`:

$$\frac{\{A\}\ c\ ; \texttt{while}\ b\ \texttt{do}\ c\ \{B\}}{\{A\}\ \texttt{do}\ c\ \texttt{while}\ b\ \{B\}}.$$

2 4F-2 VCGen for Let

- 0 pts Correct

Exercise 4F-2

From previous homework, we know that the command `let` $x = e$ `in` c is equivalent to the sequence of commands $x' := x; x := e; c; x := x'$ where x' is a fresh variable. So, we can compute the verification condition of `let` in terms of the verification conditions of these other commands:

$$\begin{aligned} VC(\texttt{let } x = e \texttt{ in } c, B) &= VC(x' := x; x := e; c; x := x', B) \\ &= VC(x' := x, VC(x := e, VC(c, VC(x := x', B)))) \\ &= [x/x'][e/x]VC(c, [x'/x]B). \end{aligned}$$

Exercise 4F-3

Let c be the command `let` $x = 2$ `in skip`, let B be the condition $x = 2$, let σ be the state $(x \mapsto 1)$. Then, based on the buggy rule for let, we have

$$\begin{aligned} VC(c, B) &= VC(\texttt{let } x = 2 \texttt{ in skip}, \text{``}x = 2\text{''}) \\ &= [2/x]VC(\texttt{skip}, \text{``}x = 2\text{''}) \\ &= [2/x]\text{``}x = 2\text{''} \\ &= \text{``}2 = 2\text{''}. \end{aligned}$$

Clearly $\sigma \vDash VC(c, B)$, because "$2 = 2$" is true regardless of the value of any variable. However, based on our operational semantics rules for `let`, we have $\langle c, \sigma \rangle \Downarrow \sigma$. That is, `let` $x = 2$ `in skip` doesn't change σ because `skip` doesn't change σ.

Finally we have $\sigma \nvDash$ "$x = 2$" because, as defined, $\sigma(x) = 1$.

Exercise 4F-4

We claim that `do` c `while` b is equivalent to executing c once and then running c in a normal `while` loop (i.e. `do` c `while` $b \sim c$; `while` b `do` c). To see this, it suffices to show that they both execute c the same number of times. But this is true because both have the behavior that, after one execution of c they check b and continue looping if and only if b was true. Using this, we can easily write a Hoare rule for `do-while` in terms of the Hoare rules we already know for sequencing and `while`:

$$\frac{\{A\}\ c\ ;\texttt{while}\ b\ \texttt{do}\ c\ \{B\}}{\{A\}\ \texttt{do}\ c\ \texttt{while}\ b\ \{B\}}.$$

3 4F-3 VCGen Mistakes

- 0 pts Correct

Exercise 4F-2

From previous homework, we know that the command let $x = e$ in c is equivalent to the sequence of commands $x' := x; x := e; c; x := x'$ where x' is a fresh variable. So, we can compute the verification condition of let in terms of the verification conditions of these other commands:

$$\begin{aligned} VC(\texttt{let } x = e \texttt{ in } c, B) &= VC(x' := x; x := e; c; x := x', B) \\ &= VC(x' := x, VC(x := e, VC(c, VC(x := x', B)))) \\ &= [x/x'][e/x]VC(c, [x'/x]B). \end{aligned}$$

Exercise 4F-3

Let c be the command let $x = 2$ in skip, let B be the condition $x = 2$, let σ be the state $(x \mapsto 1)$. Then, based on the buggy rule for let, we have

$$\begin{aligned} VC(c, B) &= VC(\texttt{let } x = 2 \texttt{ in skip}, \text{``}x = 2\text{''}) \\ &= [2/x]VC(\texttt{skip}, \text{``}x = 2\text{''}) \\ &= [2/x]\text{``}x = 2\text{''} \\ &= \text{``}2 = 2\text{''}. \end{aligned}$$

Clearly $\sigma \vDash VC(c, B)$, because "$2 = 2$" is true regardless of the value of any variable. However, based on our operational semantics rules for let, we have $\langle c, \sigma \rangle \Downarrow \sigma$. That is, let $x = 2$ in skip doesn't change σ because skip doesn't change σ.

Finally we have $\sigma \nvDash$ "$x = 2$" because, as defined, $\sigma(x) = 1$.

Exercise 4F-4

We claim that do c while b is equivalent to executing c once and then running c in a normal while loop (i.e. do c while $b \sim c$; while b do c). To see this, it suffices to show that they both execute c the same number of times. But this is true because both have the behavior that, after one execution of c they check b and continue looping if and only if b was true. Using this, we can easily write a Hoare rule for do-while in terms of the Hoare rules we already know for sequencing and while:

$$\frac{\{A\}\ c\,;\texttt{while}\ b\ \texttt{do}\ c\ \{B\}}{\{A\}\ \texttt{do}\ c\ \texttt{while}\ b\ \{B\}}.$$

4 4F-4 Axiomatic Do-While

- 0 pts Correct

Peer Review ID: 70889148 — enter this when you fill out your peer evaluation via gradescope