CS536 Science of Programming
Fall 2024
Assignment 3 Sample Solution Sketches

1. (a) $M(S,\sigma) = \big\{\{x=2,y=1\},\{x=3,y=2\},\{x=0,y=1\},\{x=3,y=1\}\big\}$

   (b) Here, I show the collection of states after each iteration.

$$\begin{aligned}
& M(W,\sigma) \\
= \;& M(W,\big\{\{x=2,y=1\},\{x=3,y=2\},\{x=0,y=1\},\{x=3,y=1\}\big\}) \\
= \;& M(W,\big\{\{x=1,y=1\},\{x=2,y=2\},\{x=3,y=3\},\{x=0,y=1\},\perp_d\big\}) \\
= \;& M(W,\big\{\{x=1,y=1\},\{x=1,y=2\},\{x=3,y=3\},\{x=0,y=1\},\perp_d\big\}) \\
= \;& \big\{\{x=1,y=1\},\{x=1,y=2\},\{x=3,y=3\},\{x=0,y=1\},\perp_d\big\}
\end{aligned}$$

2.

$$\begin{aligned}
MAJORITY \equiv\; & k_0 = 0; k_1 = 0; \\
& \textbf{while } k_0 < n \wedge k_1 < n \textbf{ do } J; k_0 := k_0 + 1; k_1 := k_1 + 1 \textbf{ od}; \\
& \textbf{if } k_1 = n \to major := 0 \;\square\; k_0 = n \to major := 1 \textbf{ fi}
\end{aligned}$$

Here, $J \equiv$ **do** $b[k_0] = 1 \to k_0 := k_0 + 1 \;\square\; b[k_1] = 0 \to k_1 := k_1 + 1$ **od**.

After the deterministic while loop, we must have $k_0 = n$ or $k_1 = n$, so there will not be any runtime error in the execution of the nondeterministic conditional statement. Actually, I think k_0 and k_1 cannot equal to n at the same time, so a deterministic conditional statement can be used here as well.

There are (should be?) other ways to implement this program. The key is to pair up one 0 and one 1 in each iteration without missing any possible pairs in one (or several) scan.

3. (a) False. A nondeterministic program can terminate in one state.

   (b) False. if $\sigma \not\models p$ then $\sigma \models \{p\}S\{q\}$.

   (c) False. If $\sigma \not\models_{tot} \{p\}S\{q\}$ then $\sigma \models p$ and $M(S,\sigma) \not\models q$.

   (d) False. If $\sigma \not\models p$, we have $\sigma \models \{p\}S\{q\}$; then we do not know anything about $M(S,\sigma)$.

   (e) True. If $\sigma \not\models \{p\}S\{q\}$, then $\sigma \models p$ and $M(S,\sigma) - \perp \not\models q$, which implies that $M(S,\sigma) \not\models q$ and thus $\sigma \not\models_{tot} \{p\}S\{q\}$.

4. (a) True, it follows the backward assignment rule immediately. Technically speaking, the backward assignment can create a partially correct triple, but in this question the statement and the post-condition are both "safe" (aka, they will not cause divergence or run-time error during evaluation), so the triple is totally correct as well.

   (b) False. A witness can be $\sigma = \{s = 0, k = 0\}$. We have $\sigma \models P(0,0)$, but $M(s := s+1, \sigma) = \{s = 1, k = 0\} \not\models P(0,1)$.

(c) True. The precondition cannot be satisfied by any state.

(d) True. The precondition logically implies $P(k, s) \wedge P(k, s_0)$. The program does not update s_0, so $P(k, s_0)$ in the post-condition is still true.

(e) True, using backward assignment we can get (both partial and totally) valid triples: $\{P(k+1, s)\}k := k+1\{P(k, s)\}$ and $\{P(k+1, s+1)\}s := s+1\{P(k+1, s)\}$. Combine them using the sequence rule and we get: $\{P(k+1, s+1)\}s := s+1; k := k+1\{P(k, s)\}$.

5. (a)
   - When $\sigma(x) = 0$, the precondition is not satisfied so the triple is satisfied.
   - When $\sigma(x) < 0$ or when $\sigma(x)$ is odd, S will diverge; which is acceptable for partial correctness.
   - When $\sigma(x) > 0$ and $\sigma(x)$ is even, S will terminate with some state τ with $\tau(x) = 0$, which does not satisfy the post-condition.
   - To sum up, for satisfaction under partial correctness, we have $\sigma(x) \leq 0$ or $\sigma(x)\%2 = 1$.

   (b) For total correctness, S has to terminate in σ; so the only possible value for x in σ is 0.

6. (a) Yes. If $\sigma \models p_1 \wedge p_2$, then $\sigma \models p_1$ and $\sigma \models p_2$. Then, $M(S, \sigma) \models q_1$ and $M(S, \sigma) \models q_2$; which logically implies that $M(S, \sigma) \models q_1 \wedge q_2$ and then $M(S, \sigma) \models q_1 \vee q_2$. Thus, $\sigma \models_{tot} \{p_1 \wedge p_2\}S\{q_1 \vee q_2\}$.

   (b) No. If $\sigma \models p_1 \vee p_2$, then $\sigma \models p_1$ or $\sigma \models p_2$.
   - If $\sigma \models p_1$ and $\sigma \models p_2$, then $M(S, \sigma) \models q_1$ and $M(S, \sigma) \models q_2$; which logically implies that $M(S, \sigma) \models q_1 \wedge q_2$.
   - If $\sigma \models p_1$ and $\sigma \not\models p_2$, then we only know $M(S, \sigma) \models q_1$ and we do not have any information to decide whether $M(S, \sigma) \models q_2$. It is a similar case when $\sigma \not\models p_1$ and $\sigma \models p_2$.

   (c) Yes. If $\sigma \models p_1 \vee p_2$, then $\sigma \models p_1$ or $\sigma \models p_2$. Then, $M(S, \sigma) \models q_1$ or $M(S, \sigma) \models q_2$; which logically implies that $M(S, \sigma) \models q_1 \vee q_2$. Thus, $\sigma \models_{tot} \{p_1 \vee p_2\}S\{q_1 \vee q_2\}$.

7. (a) True. For any state σ, if $\sigma \models p_1 \wedge p_2$, then $\sigma \models p_1$ and $\sigma \models p_2$. Then, $M(S, \sigma) - \perp \models q_1$ and $M(S, \sigma) - \perp \models q_2$; which logically implies that $M(S, \sigma) - \perp \models q_1 \wedge q_2$. Thus, $\models \{p_1 \wedge p_2\}S\{q_1 \vee q_2\}$.

   (b) True. The post-condition of the triple semantically equals to $\neg q_1 \vee q_2$, and $q_2 \Rightarrow \neg q_1 \vee q_2$. Since $\models \{p_2\}S\{q_2\}$, thus $\models \{p_2\}S\{q_1 \rightarrow q_2\}$.

   (c) True. The triple semantically equals to $\{p_1 \vee p_2\}S\{q_1 \vee q_2\}$. Using a proof similar to question 7(a) we can prove that this triple is valid under partial correctness.

8. (a) True. The validity of the triple follows from the definition of $wp(S, q)$ immediately.

   (b) True. The definition of $wp(S, q)$ implies $\models_{tot} \{w\}S\{q\}$, which also implies $\models \{w\}S\{q\}$. By strengthening the precondition from w to $w \wedge q$, we can get the valid triple in the question.

   (c) False. The definition of $wp(S, q)$ logically implies that if $\sigma \not\models wp(S, q)$, then $M(S, \sigma) \not\models q$.

(d) True. It follows from $\models_{tot} \{w\}S\{q\}$.

(e) True. For any state $\sigma \models \neg w$, $M(S, \sigma)$ must be either a pseudo-state or a state satisfies $\neg q$. Thus, if $\sigma \not\models w$ we can get $\sigma \models \{\neg w\}S\{\neg q\}$.

9. (a)

$$wlp(S, q) \equiv wlp(y := y\%x,\ sqrt(y) > x) \equiv sqrt(y\%x) > x$$

(b)

$$\begin{aligned} D(S) &\equiv D(y := y\%x) \Leftrightarrow x \neq 0 \\ D(wlp(S, q)) &\equiv D(sqrt(y\%x) > x) \Leftrightarrow y\%x \geq 0 \wedge x \neq 0 \end{aligned}$$

Thus,

$$\begin{aligned} wp(S, q) &\equiv (x \neq 0) \wedge (sqrt(y\%x) > x) \wedge (y\%x \geq 0 \wedge x \neq 0) \\ &\Leftrightarrow x \neq 0 \wedge sqrt(y\%x) > x \wedge y\%x \geq 0 \end{aligned}$$

10. (a)

$$\begin{aligned} wlp(S, q) &\equiv wlp(\textbf{if } y \geq 0 \to x := y/x \ \Box\ x \geq 0 \to x := x/y \textbf{ fi}, x < y < z) \\ &\equiv (y \geq 0 \to wlp(x := y/x, x < y < z)) \\ &\qquad \wedge (x \geq 0 \to wlp(x := x/y, x < y < z)) \\ &\equiv (y \geq 0 \to y/x < y < z) \wedge (x \geq 0 \to x/y < y < z) \end{aligned}$$

(b)

$$\begin{aligned} D(S) &\equiv D(y \geq 0 \vee x \geq 0) \wedge (x \geq 0 \vee y \geq 0) \wedge (y \geq 0 \to D(x := y/x)) \\ &\qquad \wedge (y < 0 \to D(x := x/y)) \\ &\Leftrightarrow (x \geq 0 \vee y \geq 0) \wedge (y \geq 0 \to x \neq 0) \wedge (x \geq 0 \to y \neq 0) \\ D(wlp(S, q)) &\equiv D\big((y \geq 0 \to y/x < y < z) \wedge (x \geq 0 \to x/y < y < z)\big) \\ &\Leftrightarrow x \neq 0 \wedge y \neq 0 \end{aligned}$$

Thus,

$$\begin{aligned} wp(S, q) \equiv\ & (x \geq 0 \vee y \geq 0) \wedge (y \geq 0 \to x \neq 0) \wedge (x \geq 0 \to y \neq 0) \\ & (y \geq 0 \to y/x < y < z) \wedge (x \geq 0 \to x/y < y < z) \wedge (x \neq 0 \wedge y \neq 0) \end{aligned}$$