CS536 Science of Programming
Fall 2024
Assignment 2 Sample Solution Sketches

1. (a) $\sigma \models \exists x \in S.p$ means for **this** state σ and for **some** $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha] \models p$.

   (b) $\sigma \models \forall x \in S.p$ means for **this** state σ and for **all** $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha] \models p$.

   (c) $\sigma \not\models \exists x \in S.p$ means for **this** state σ and for **all** $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha] \not\models p$.

   (d) $\sigma \not\models \forall x \in S.p$ means for **this** state σ and for **some** $\alpha \in S$, it is the case that $\sigma[x \mapsto \alpha] \not\models p$.

   (e) $\models \exists x \in S.p$ means for **all** state σ, it is the case that $\sigma \models \exists x \in S.p$.

   (f) $\models \forall x \in S.p$ means for **all** state σ, it is the case that $\sigma \models \forall x \in S.p$.

   (g) $\not\models \exists x \in S.p$ means for **some** state σ, it is the case that $\sigma \not\models \exists x \in S.p$.

   (h) $\not\models \forall x \in S.p$ means for **some** state σ, it is the case that $\sigma \not\models \forall x \in S.p$.

2. (a) True. $\{x = 2, y = 3\}(x < 2) = F$, and by the absurdity rule, False implies any predicate is True.

   (b) False. Denote $\beta = (2, 5, 4, 8)$, the sequence of values bound with b. For all m between 0 and 3, we have $\beta(m) \geq 2$.

   (c) False. If there exists such y, then $b[0]$, $b[1]$ and y are all evaluated to the same value in the given state; however we can see that $b[0], b[1]$ are bound with different values in the given state.

   (d) False. For witness, we can find value 100 for x and 0 for k, then $\{x = 1, b = (5, 3, 6)\}[x \mapsto 100][k \mapsto 0](x < b[k]) = (100 < 5) = F$.

3. (a) $i := 0;$ **while** $i < size(b)$ **do** $b[i] := i;\ i := i + 1$ **od**

   (b) **while** $x \neq 1$ **do** **if** $x\%2 = 0$ **then** $x := x/2$ **else** $x := x + 1$ **fi** **od**

   (c) $m := 8;\ p := 1;\ y := 1;\ m := m+1;$ **while** $m < 20$ **do** $p := p * y;\ y := y+1;\ m := m + 1$ **od**

4. (a)

$$
\begin{aligned}
& \langle\ \textbf{if } x < 2 \textbf{ then } x := y + 1, w := x + 2 \textbf{ fi},\ \{x = 3, y = 3, w = 4\}\ \rangle \\
\rightarrow\ & \langle\ \textbf{skip}, \{x = 3, y = 3, w = 4\}\ \rangle \\
\rightarrow\ & \langle\ E, \{x = 3, y = 3, w = 4\}\ \rangle
\end{aligned}
$$

(b) Let us denote $W \equiv$ **while** $x < 2$ **do** $x := y + 1, w := x + 2$ **od** .

$$
\begin{array}{ll}
 & \langle\ W, \{x = 1, y = 3, w = 4\}\ \rangle \\
\to & \langle\ x := y + 1, w := x + 2; W, \{x = 1, y = 3, w = 4\}\ \rangle \\
\to & \langle\ w := x + 2; W, \{x = 4, y = 3, w = 4\}\ \rangle \\
\to & \langle\ W, \{x = 4, y = 3, w = 6\}\ \rangle \\
\to & \langle\ E, \{x = 4, y = 3, w = 6\}\ \rangle
\end{array}
$$

(c)

$$
\begin{array}{ll}
 & \langle\ x := y + 1; y := x + 1, \sigma\ \rangle \\
\to & \langle\ y := x + 1, \sigma[x \mapsto \sigma(y) + 1]\ \rangle \\
\to & \langle\ E, \sigma[x \mapsto \sigma(y) + 1]\big[y \mapsto \sigma[x \mapsto \sigma(y) + 1](x) + 1\big]\ \rangle \\
= & \langle\ E, \sigma[x \mapsto \sigma(y) + 1][y \mapsto \sigma(y) + 2]\ \rangle
\end{array}
$$

5. (a)

$$
\begin{array}{lll}
 & \langle\ W, \sigma_1\ \rangle & \\
\to & \langle\ S; W, \sigma_1\ \rangle & \text{because } \sigma_1(x) > \sigma_1(y) \\
\to & \langle\ y := -2 * x; W, \sigma_1\ \rangle & \text{because } \sigma_1(x) \leq 0 \\
\to & \langle\ W, \sigma_1[y \mapsto -2 * \sigma_1(x)]\ \rangle & \\
\to & \langle\ E, \sigma_1[y \mapsto -2 * \sigma_1(x)]\ \rangle & \text{because } \sigma(x) \leq 0 \text{ and } \sigma_1[y \mapsto -2 * \sigma_1(x)](y) \geq 0
\end{array}
$$

(b)

$$
\begin{array}{lll}
 & \langle\ W, \sigma_2\ \rangle & \\
\to^* & \langle\ W, \sigma_2[x \mapsto \sigma_2(x) + 1]\ \rangle & \text{because } \sigma_2(x) > \sigma_2(y) \text{ and } \sigma_2(x) > 0 \\
\to^* & \langle\ W, \sigma_2[x \mapsto \sigma_2(x) + 2]\ \rangle & \\
\to^* & \langle\ E, \perp_d\ \rangle & \text{the evaluation of } x - y \text{ can only get larger}
\end{array}
$$

6. (a)

$$
\begin{array}{ll}
 & M(S, \tau) \\
= & M(y := y * x, \tau[x \mapsto \tau(x) + 1]) \\
= & M(E, \tau[x \mapsto \tau(x) + 1][y \mapsto \tau[x \mapsto \tau(x) + 1](y * x)]) \\
= & \{\tau[x \mapsto \tau(x) + 1][y \mapsto \tau[x \mapsto \tau(x) + 1](y * x)]\} \\
= & \{\tau[x \mapsto \tau(x) + 1][y \mapsto \tau(y) * (\tau(x) + 1)]\}
\end{array}
$$

(b) $M(W, \sigma) = \{\sigma\}$, because $\sigma(x < 3) = F$.

(c)

$$\begin{aligned}
& M(W, \sigma) & \\
= \ & M(x := x+1; y := y*x; W,\ \sigma) & \text{because } \sigma(x<3)=T \\
= \ & M(y := y*x; W,\ \sigma[x \mapsto 2]) & \\
= \ & M(W,\ \sigma[x \mapsto 2][y \mapsto 2]) & \\
= \ & M(W,\ \sigma[x \mapsto 3][y \mapsto 6]) & \text{after another iteration} \\
= \ & \{\sigma[x \mapsto 3][y \mapsto 6]\} &
\end{aligned}$$

7. (a) $M(S, \sigma) = M(x := y/x, \sigma) = M(E, \sigma[x \mapsto \sigma(y/x)]) = \{\sigma[x \mapsto 0]\}$

   (b)

$$\begin{aligned}
& M(W, \sigma) & \sigma = \{x=2, y=2, b=(0,1,2)\} \\
= \ & M(S; W, \sigma\}) & \sigma(x>0)=T \\
= \ & M(x := x-1; y := b[y]; W, \sigma\}) & \sigma(x<y)=F \\
= \ & M(W, \sigma[x \mapsto 1][y \mapsto 2]\}) & \\
= \ & M(S; W, \sigma[x \mapsto 1][y \mapsto 2]\}) & \sigma[x \mapsto 1][y \mapsto 2](x>0)=T \\
= \ & M(x := y/x; W, \sigma[x \mapsto 1][y \mapsto 2]\}) & \sigma[x \mapsto 1][y \mapsto 2](x<y)=T \\
= \ & M(W, \{x=2, y=2, b=(0,1,2)\}) & \\
= \ & \{\perp_d\} & \text{the same configuration appears again}
\end{aligned}$$

   (c)

$$\begin{aligned}
& M(W, \sigma) & \\
= \ & M(W, \{x=8, y=2, b=(4,2,0)\}) & \text{after the first iteration} \\
= \ & M(W, \{x=7, y=0, b=(4,2,0)\}) & \text{after the second iteration} \\
= \ & M(W, \{x=6, y=4, b=(4,2,0)\}) & \text{after another iteration} \\
= \ & \{\perp_e\} & \text{array index out of bounds}
\end{aligned}$$

   (d) There is no such state. The only division operation y/x appears in the if condition of statement S, and we evaluate y/x immediately after we enters an iteration of W with $x < y$ evaluated to $True$. Thus, whenever we evaluate y/x, we must have the evaluation of $x > 0$ equals $True$.

8. To have $M(S, \sigma) = \{\perp_e\}$, we need state σ with:

   - either $\sigma(x) < 0$, because of a “square root of negative number” error;
   - or $\sigma(y) < 0$ or $\sigma(y) \geq 4$, because of a “array index out of bound” error;
   - or $\sigma(y) = 1$, because of a “division by 0” error.

9. (a) False. A pseudo state does not satisfy any predicate, even if the predicate is valid.

   (b) True. A pseudo state does not satisfy any predicate.

   (c) False. If $\sigma(p) \neq \perp$, then we can say $\sigma(p) = T$ or $\sigma(p) = F$, we do not necessarily have $\sigma \models p$.

(d) True. If $\sigma(p) = \perp$, then $\sigma(\neg p) = \perp$; thus, there exists some state, aka σ, that does not satisfy $\neg p$.

(e) True. If $\models p$, then all states (that are proper for p) satisfy p; in other words, there does not exist any state (that is proper for p) that does not satisfy p.

10. (a) True. Since $\Sigma_0 \models p$, then $\forall \sigma \in \Sigma_0.\ \sigma \models p$. We also have $\tau \models p$, so $\forall \sigma \in \Sigma_0 \cup \{\tau\}.\ \sigma \models p$; thus, $\Sigma_0 \cup \{\tau\} \models p$.

(b) True. No matter what p is, we always have that each state in $\emptyset$ satisfies p, since there are no states in $\emptyset$.

(c) False. It is possible that $\tau(p) = \perp$.

(d) False. For any state $\tau \in \Sigma_0$, it is possible that $\tau(p) = \perp$. Even if every state τ can evaluate p to T or F, the statements is still not true, since we might have some states in Σ_0 satisfy p and the others satisfy $\neg p$.

(e) True. Since $\sigma_1 \models p_1$ and $\sigma_1(p_2) \neq \perp$, thus $\sigma_1 \models p_1 \vee p_2$. A similar statement also holds for σ_2, so $\{\sigma_1, \sigma_2\} \models p_1 \vee p_2$.