CS536 Science of Programming
Fall 2024
Assignment 4 Sample Solution Sketches

1. (a)

$$\begin{aligned} p[y+z/x] &\equiv \Big(w * x \neq 0 \land z \leq 2 \to f(w) > 0 \land \forall x.\ \exists y.\ 0 \leq y \leq x \\ &\quad \land f(w \div x) + y > f(z)\Big)[y+z/x] \\ &\equiv w * (y+z) \neq 0 \land z \leq 2 \to f(w) > 0 \land \forall x.\ \exists y.\ 0 \leq y \leq x \\ &\quad \land f(w \div x) + y > f(z) \end{aligned}$$

(b)

$$\begin{aligned} p[x+z/w] &\equiv \Big(w * x \neq 0 \land z \leq 2 \to f(w) > 0 \land \forall x.\ \exists y.\ 0 \leq y \leq x \\ &\quad \land f(w \div x) + y > f(z)\Big)[x+z/w] \\ &\equiv (x+z) * x \neq 0 \land z \leq 2 \to f(x+z) > 0 \land \Big(\forall x_0.\ \exists y.\ 0 \leq y \leq x_0 \\ &\quad \land f(w \div x_0) + y > f(z)\Big)[x+z/w] \\ &\equiv (x+z) * x \neq 0 \land z \leq 2 \to f(x+z) > 0 \land \forall x_0.\ \exists y.\ 0 \leq y \leq x_0 \\ &\quad \land f\big((x+z) \div x_0\big) + y > f(z) \end{aligned}$$

(c)

$$\begin{aligned} p[x+y/z] &\equiv \Big(w * x \neq 0 \land z \leq 2 \to f(w) > 0 \land \forall x.\ \exists y.\ 0 \leq y \leq x \\ &\quad \land f(w \div x) + y > f(z)\Big)[x+y/z] \\ &\equiv w * x \neq 0 \land (x+y) \leq 2 \to f(w) > 0 \land \Big(\forall x_0.\ \exists y_0.\ 0 \leq y_0 \leq x_0 \\ &\quad \land f(w \div x_0) + y_0 > f(z)\Big)[x+y/z] \\ &\equiv w * x \neq 0 \land (x+y) \leq 2 \to f(w) > 0 \land \forall x_0.\ \exists y_0.\ 0 \leq y_0 \leq x_0 \\ &\quad \land f(w \div x_0) + y_0 > f(x+y) \end{aligned}$$

2. (a) One of the examples can be:

$$(x * y)[2\ /\ x][4\ /\ y] \equiv (x * y)[4\ /\ y][2\ /\ x]$$

They both are syntactically equivalent to $2*4$, so they are syntactically equivalent to each other. The key here is to choose an expression e containing no y and an expression e' containing no x.

(b) One of the counterexamples can be:

$$(x * y)[y\ /\ x][2\ /\ y] \not\equiv (x * y)[2\ /\ y][y\ /\ x]$$

Because $(x * y)[y \;/\; x][2 \;/\; y] \equiv (y * y)[2 \;/\; y] \equiv 2 * 2$ and $(x * y)[2 \;/\; y][y \;/\; x] \equiv (x * 2)[y \;/\; x] \equiv y * 2$, so they are not syntactically equivalent to each other. The key here is to choose an expression e containing y and/or an expression e' containing x.

3. (a) From the definition of weakest liberal precondition, we have that $p \Leftrightarrow wlp(S, q)$ logically implies that $\models \{p\}S\{q\}$; which logically implies that $sp(p, S) \Rightarrow q$.

   (b) A counterexample can be $S \equiv x := x * x$ and $q \equiv x < 1$. We can calculate $wlp(S, q) \equiv x * x < 1 \Leftrightarrow x = 0$. But, $sp(x = 0, x := x * x) \equiv (x_0 = 0 \wedge x = x_0 * x_0) \Rightarrow x = 0$, which is strictly stronger than $x < 1$.

4. (a) False. $s \Leftrightarrow sp(p, S)$ logically implies that $\models \{p\}S\{s\}$, but not that $\models_{tot} \{p\}S\{s\}$. It is possible that $\perp \in M(S, \sigma)$ for some $\sigma \models p$.

   (b) False. $s \Leftrightarrow sp(p, S)$ logically implies that $\models \{p\}S\{s\}$; which means for all state σ, it is the case that $\sigma \models \{p\}S\{s\}$

   (c) False. For some state $\sigma \models p$, it is possible that $\perp \in M(S, \sigma)$, then $M(S, \sigma) \not\models s$.

   (d) False. Even if $M(S, \sigma) \not\models s$, it is still possible that $s \not\models p$.

   (e) False. If $\sigma \not\models p$, we do not know anything between $M(S, \sigma)$ and s.

5. Denote $IF \equiv$ **if** $x \geq 0 \rightarrow x := y + 1; z := x \;\square\; x \leq 0 \rightarrow y := x - 1; z := y$ **fi**.

$$
\begin{aligned}
& aged(x = y, IF) = \{x, y, z\} \cap \{x, y\} = \{x, y\}. \\
& sp(x = y, IF) \\
\equiv\ & sp(x = y \wedge x = x_0 \wedge y = y_0 \wedge x \geq 0,\ x := y + 1; z := x) \\
& \qquad \vee sp(x = y \wedge x = x_0 \wedge y = y_0 \wedge x \leq 0,\ y := x - 1; z := y) \\
\equiv\ & sp(x_0 = y \wedge x_0 = x_0 \wedge y = y_0 \wedge x_0 \geq 0 \wedge x = y + 1,\ z := x) \\
& \qquad \vee sp(x = y_0 \wedge x = x_0 \wedge y_0 = y_0 \wedge x \leq 0 \wedge y = x - 1,\ z := y) \\
\equiv\ & (x_0 = y \wedge x_0 = x_0 \wedge y = y_0 \wedge x_0 \geq 0 \wedge x = y + 1 \wedge z = x) \\
& \qquad \vee (x = y_0 \wedge x = x_0 \wedge y_0 = y_0 \wedge x \leq 0 \wedge y = x - 1 \wedge z = y)
\end{aligned}
$$

6.

$$
\begin{aligned}
& sp(y = x + 1,\ y := y + 1;\ \textbf{if } x < 0 \textbf{ then } y := -y \textbf{ fi}) \\
\equiv\ & sp(y_0 = x + 1 \wedge y = y_0 + 1,\ \textbf{if } x < 0 \textbf{ then } y := -y \textbf{ else skip fi}) \\
& \quad \#\ aged(y_0 = x + 1 \wedge y = y_0 + 1,\ \textbf{if } x < 0 \textbf{ then } y := -y \textbf{ else skip fi}) = \{y\} \\
\equiv\ & sp(y_0 = x + 1 \wedge y_1 = y_0 + 1 \wedge y = y_1 \wedge x < 0,\ y := -y) \\
& \qquad \vee sp(y_0 = x + 1 \wedge y_1 = y_0 + 1 \wedge y = y_1 \wedge x \geq 0,\ \textbf{skip}) \\
\equiv\ & (y_0 = x + 1 \wedge y_1 = y_0 + 1 \wedge y_1 = y_1 \wedge x < 0 \wedge y = -y_1) \\
& \qquad \vee (y_0 = x + 1 \wedge y_1 = y_0 + 1 \wedge y = y_1 \wedge x \geq 0)
\end{aligned}
$$

7. We can give the following formal proof.

1	$\{p \wedge B\}\ S_1\ \{q_1\}$	premise
2	$\{p \wedge \neg B\}\ S_2\ \{q_2\}$	premise
3	$\{(B \rightarrow p \wedge B) \wedge (\neg B \rightarrow p \wedge \neg B)\}$ **if** B **then** S_1 **else** S_2 **fi** $\{q_1 \vee q_2\}$	if-else 1,2
4	$(B \rightarrow p \wedge B) \wedge (\neg B \rightarrow p \wedge \neg B) \Leftrightarrow \big(B \wedge (p \wedge B)\big) \vee \big(\neg B \wedge (p \wedge \neg B)\big)$	predicate logic
5	$\big(B \wedge (p \wedge B)\big) \vee \big(\neg B \wedge (p \wedge \neg B)\big) \Leftrightarrow p$	predicate logic
	# $LHS \Leftrightarrow p \wedge B \vee p \wedge \neg B \Leftrightarrow p \wedge (B \vee \neg B) \Leftrightarrow p$	
6	$\{p\}$ **if** B **then** S_1 **else** S_2 **fi** $\{q_1 \vee q_2\}$	strengthen precondition 5,3

8.

$$wlp(x := x * x;\ y := 2 * y,\ x = y) \equiv wlp(x := x * x,\ x = 2 * y) \equiv x * x = 2 * y$$

We can give the following formal proof.

1	$\{x = 2 * y\}\ y := 2 * y\ \{x = y\}$	backward assignment
2	$\{x * x = 2 * y\}\ x := x * x\ \{x = 2 * y\}$	backward assignment
3	$\{x * x = 2 * y\}\ x := x * x;\ y := 2 * y\ \{x = y\}$	sequence 2,1

9. $p_1\ :\ x = 2^k \wedge k < n$

   $p_2\ :\ x_0 = 2^k \wedge k < n \wedge x = x_0 * 2$

   R_1 : forward assignment

   $p_3\ :\ x_0 = 2^{k_0} \wedge k_0 < n \wedge x = x_0 * 2 \wedge k = k_0 + 1$

   R_2 : forward assignment

   R_3 : sequence 1,2

   R_4 : strengthen precondition 3,4

   $p_4\ :\ p \wedge k \geq n$

   R_5 : loop 5

10. R_1 : forward assignment

    R_2 : forward assignment

    R_3 : weaken postcondition 2,3

    R_4 : sequence 1,4

    R_5 : backward assignment

    R_6 : backward assignment

    R_7 : strengthen precondition 8,7

    R_8 : sequence 9,6

    R_9 : loop 10

    R_{10} : sequence 5,11

    R_{11} : weaken postcondition 12,13