CS536 Science of Programming
Fall 2024
Assignment 6 Sample Solution Sketches

1. (a) The reasonable pre- and post-conditions here can be $n \geq 0$ and $x = fac(n)$.
   (b) A reasonable loop condition suggested in the question is $p \equiv x = fac(y) \wedge 0 \leq y \leq n$. Since the program will end with $y = n$, so the loop condition should be $B \equiv y \neq n$.
   (c) We run the loop starting with $y = 0$ (since this is the other boundary of the range of y) and we will increase the value of y in each iteration, which means we can include "$-y$" in the bound expression. And $n - y$ is a good loop bound expression, since the loop invariant implies that $n - y \geq 0$.

2. Here is one possible full proof outline for this question.

   $\{n \geq 0\}$
   $x := 1; \{n \geq 0 \wedge x = 1\}\ y := 0; \{n \geq 0 \wedge x = 1 \wedge y = 0\}$
   $\{\textbf{inv}\ p \equiv x = fac(y) \wedge 0 \leq y \leq n\}\{\textbf{bd}\ n - y\}$
   $\textbf{while}\ y \neq n\ \textbf{do}$
   $\quad \{x = fac(y) \wedge 0 \leq y \leq n \wedge y \neq n \wedge n - y = t_0\}$
   $\quad \{x * (y+1) = fac(y+1) \wedge 0 \leq (y+1) \leq n \wedge n - (y+1) < t_0\}\ x := x * (y+1);$
   $\quad \{x = fac(y+1) \wedge 0 \leq (y+1) \leq n \wedge n - (y+1) < t_0\}\ y := y + 1$
   $\quad \{x = fac(y) \wedge 0 \leq y \leq n \wedge n - y < t_0\}$
   $\textbf{od}$
   $\{x = fac(y) \wedge 0 \leq y \leq n \wedge y = n\}$
   $\{x = fac(n)\}$

   There are other ways to write this program. For example, you might write the base case and the loop in an $if - else$ statement; you might give x and y different initial values; your loop body might be different ...

3. We can find p using Backward Assignment and we can create the following full proof outline.

   $$\{p\}\ b[i] := b[j]; \{p_1\}\ b[j] := b[k]\ \{b[i] > b[k]\}$$

   Here,

   $$\begin{aligned} p_1 &\equiv (b[i] > b[k])\big[b[k]/b[j]\big] \\ &\equiv (\textbf{if}\ i = j\ \textbf{then}\ b[k]\ \textbf{else}\ b[i]\ \textbf{fi}) > (\textbf{if}\ k = j\ \textbf{then}\ b[k]\ \textbf{else}\ b[k]\ \textbf{fi}) \\ &\mapsto (\textbf{if}\ i = j\ \textbf{then}\ b[k]\ \textbf{else}\ b[i]\ \textbf{fi}) > b[k] \\ &\mapsto \textbf{if}\ i = j\ \textbf{then}\ b[k] > b[k]\ \textbf{else}\ b[i] > b[k]\ \textbf{fi} \\ &\mapsto \textbf{if}\ i = j\ \textbf{then}\ F\ \textbf{else}\ b[i] > b[k]\ \textbf{fi} \\ &\mapsto i \neq j \wedge b[i] > b[k] \end{aligned}$$

and,

$$\begin{aligned}
p &\equiv (i \neq j \wedge b[i] > b[k])\big[b[j]/b[i]\big] \\
&\equiv i \neq j \wedge b[j] > (\textbf{if } k = i \textbf{ then } b[j] \textbf{ else } b[k] \textbf{ fi}) \\
&\mapsto i \neq j \wedge (\textbf{if } k = i \textbf{ then } b[j] > b[j] \textbf{ else } b[j] > b[k] \textbf{ fi}) \\
&\mapsto i \neq j \wedge (\textbf{if } k = i \textbf{ then } F \textbf{ else } b[j] > b[k] \textbf{ fi}) \\
&\mapsto i \neq j \wedge k \neq i \wedge b[j] > b[k]
\end{aligned}$$

4. We can use Backward Assignment to create the following full proof outline.

$$\{k < b[k] < b[j]\}\{p\}\ b\big[b[k]\big] := b[j]\ \{b[k] \neq b[j]\}$$

Here,

$$\begin{aligned}
p &\equiv (b[k] \neq b[j])\ \big[b[j]/b\big[b[k]\big]\big] \\
&\equiv (\textbf{if } k = b[k] \textbf{ then } b[j] \textbf{ else } b[k] \textbf{ fi}) \neq (\textbf{if } j = b[k] \textbf{ then } b[j] \textbf{ else } b[j] \textbf{ fi}) \\
&\mapsto (\textbf{if } k = b[k] \textbf{ then } b[j] \textbf{ else } b[k] \textbf{ fi}) \neq b[j] \\
&\mapsto \textbf{if } k = b[k] \textbf{ then } b[j] \neq b[j] \textbf{ else } b[k] \neq b[j] \textbf{ fi} \\
&\mapsto \textbf{if } k = b[k] \textbf{ then } F \textbf{ else } b[k] \neq b[j] \textbf{ fi} \\
&\mapsto k \neq b[k] \wedge b[k] \neq b[j]
\end{aligned}$$

It is easy to see that $k < b[k] < b[j] \Rightarrow k \neq b[k] \wedge b[k] \neq b[j]$, and we finish the proof outline.

5. Here is an evaluation graph for configuration $\langle [x := 1 \parallel x := -1]; y := y + x,\ \sigma \rangle$.

$$\begin{tikzcd}
& \langle [x := 1 \parallel x := -1]; y := y + x,\ \sigma \rangle \arrow[dl] \arrow[dr] & \\
\langle [E \parallel x := -1]; y := y + x,\ \sigma[x \mapsto 1] \rangle \arrow[d] & & \langle [x := 1 \parallel E]; y := y + x,\ \sigma[x \mapsto -1] \rangle \arrow[d] \\
\langle [E \parallel E]; y := y + x,\ \sigma[x \mapsto -1] \rangle \arrow[d] & & \langle [E \parallel E]; y := y + x,\ \sigma[x \mapsto 1] \rangle \arrow[d] \\
\langle E,\ \sigma[x \mapsto -1][y \mapsto \sigma(y) - 1] \rangle & & \langle E,\ \sigma[x \mapsto 1][y \mapsto \sigma(y) + 1] \rangle
\end{tikzcd}$$

6. Here is an evaluation graph for configuration $\langle W, \{x = 0, y = 1, n = 2\}\rangle$ where $W \equiv \textbf{while } x < n \textbf{ do } [x := x + 1 \parallel y := y * 2] \textbf{ od}$.

$$
\begin{tikzcd}[column sep=tiny]
 & \langle \textbf{while } x < n \textbf{ do } [x := x + 1 \parallel y := y * 2] \textbf{ od},\ \{x = 0, y = 1, n = 2\}\rangle \arrow[d] & \\
 & \langle [x := x + 1 \parallel y := y * 2]; W,\ \{x = 0, y = 1, n = 2\}\rangle \arrow[dl] \arrow[dr] & \\
\langle [E \parallel y := y * 2]; W,\ \{x = 1, y = 1, n = 2\}\rangle \arrow[dr] & & \langle [x := x + 1 \parallel E]; W,\ \{x = 0, y = 2, n = 2\}\rangle \arrow[dl] \\
 & \langle W,\ \{x = 1, y = 2, n = 2\}\rangle \arrow[d] & \\
 & \langle [x := x + 1 \parallel y := y * 2]; W,\ \{x = 1, y = 2, n = 2\}\rangle \arrow[dl] \arrow[dr] & \\
\langle [E \parallel y := y * 2]; W,\ \{x = 2, y = 2, n = 2\}\rangle \arrow[dr] & & \langle [x := x + 1 \parallel E]; W,\ \{x = 1, y = 4, n = 2\}\rangle \arrow[dl] \\
 & \langle W,\ \{x = 2, y = 4, n = 2\}\rangle \arrow[d] & \\
 & \langle E,\ \{x = 2, y = 4, n = 2\}\rangle &
\end{tikzcd}
$$

7. We can create the following tests for disjointedness and disjoint conditions.

i	j	$change(S_i)$	$vars(S_j)$	$free(p_j, q_j)$	S_i *int* S_j ?	S_i *int* $cond_j$?
1	2	y	z	x, z	*No*	*No*
2	1	z	x, y	x, y	*No*	*No*

From the test results, we can see that:

(a) These two threads are disjoint.

(b) These two threads have disjoint conditions.

8. (a) To test whether S_1^* interferes with S_2^*, we need to test to validity of the following triples:

   - $\{p_2 \wedge q_1\} \ < T_1 > \{q_1\}$
   - $\{p_2 \wedge q_3\} \ < T_1 > \{q_3\}$
   - $\{p_2 \wedge q_4\} \ < T_1 > \{q_4\}$
   - $\{p_3 \wedge q_1\}$ **skip** $\{q_1\}$
   - $\{p_3 \wedge q_3\}$ **skip** $\{q_3\}$
   - $\{p_3 \wedge q_4\}$ **skip** $\{q_4\}$

   As an aside, the last three triples are trivially valid.

   (b) To test whether S_2^* interferes with S_1^*, we need to test to validity of the following triples:

   - $\{q_1 \wedge p_1\} \ < T_2 > \{p_1\}$
   - $\{q_1 \wedge p_2\} \ < T_2 > \{p_2\}$
   - $\{q_1 \wedge p_3\} \ < T_2 > \{p_3\}$
   - $\{q_1 \wedge p_4\} \ < T_2 > \{p_4\}$
   - $\{q_3 \wedge p_1\} \ < T_3 > \{p_1\}$
   - $\{q_3 \wedge p_2\} \ < T_3 > \{p_2\}$
   - $\{q_3 \wedge p_3\} \ < T_3 > \{p_3\}$
   - $\{q_3 \wedge p_4\} \ < T_3 > \{p_4\}$