## CS536 Science of Programming Fall 2024 Assignment 6 Sample Solution Sketches

- 1. (a) The reasonable pre- and post-conditions here can be  $n \ge 0$  and x = fac(n).
  - (b) A reasonable loop condition suggested in the question is  $p \equiv x = fac(y) \land 0 \le y \le n$ . Since the program will end with y = n, so the loop condition should be  $B \equiv y \ne n$ .
  - (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  $n y \ge 0$ .
- 2. Here is one possible full proof outline for this question.

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

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,

$$p_1 \equiv (b[i] > b[k]) \big[ b[k]/b[j] \big]$$

$$\equiv (\mathbf{if} \ i = j \ \mathbf{then} \ b[k] \ \mathbf{else} \ b[i] \ \mathbf{fi}) > (\mathbf{if} \ k = j \ \mathbf{then} \ b[k] \ \mathbf{else} \ b[k] \ \mathbf{fi})$$

$$\mapsto (\mathbf{if} \ i = j \ \mathbf{then} \ b[k] \ \mathbf{else} \ b[i] \ \mathbf{fi}) > b[k]$$

$$\mapsto \mathbf{if} \ i = j \ \mathbf{then} \ b[k] > b[k] \ \mathbf{else} \ b[i] > b[k] \ \mathbf{fi}$$

$$\mapsto \mathbf{if} \ i = j \ \mathbf{then} \ F \ \mathbf{else} \ b[i] > b[k] \ \mathbf{fi}$$

$$\mapsto i \neq j \land b[i] > b[k]$$

and,

$$p \equiv (i \neq j \land b[i] > b[k]) [b[j]/b[i]]$$

$$\equiv i \neq j \land b[j] > (\mathbf{if} \ k = i \ \mathbf{then} \ b[j] \ \mathbf{else} \ b[k] \ \mathbf{fi})$$

$$\mapsto i \neq j \land (\mathbf{if} \ k = i \ \mathbf{then} \ b[j] > b[j] \ \mathbf{else} \ b[j] > b[k] \ \mathbf{fi})$$

$$\mapsto i \neq j \land (\mathbf{if} \ k = i \ \mathbf{then} \ F \ \mathbf{else} \ b[j] > b[k] \ \mathbf{fi})$$

$$\mapsto i \neq j \land k \neq i \land b[j] > b[k]$$

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

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

Here,

$$p \equiv (b[k] \neq b[j]) \ [b[j]/b[b[k]]]$$

$$\equiv (\mathbf{if} \ k = b[k] \ \mathbf{then} \ b[j] \ \mathbf{else} \ b[k] \ \mathbf{fi}) \neq (\mathbf{if} \ j = b[k] \ \mathbf{then} \ b[j] \ \mathbf{else} \ b[j] \ \mathbf{fi})$$

$$\mapsto (\mathbf{if} \ k = b[k] \ \mathbf{then} \ b[j] \ \mathbf{else} \ b[k] \ \mathbf{fi}) \neq b[j]$$

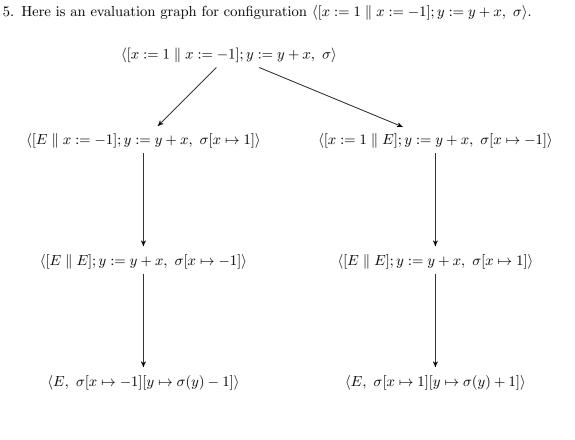
$$\mapsto \mathbf{if} \ k = b[k] \ \mathbf{then} \ b[j] \neq b[j] \ \mathbf{else} \ b[k] \neq b[j] \ \mathbf{fi}$$

$$\mapsto \mathbf{if} \ k = b[k] \ \mathbf{then} \ F \ \mathbf{else} \ b[k] \neq b[j] \ \mathbf{fi}$$

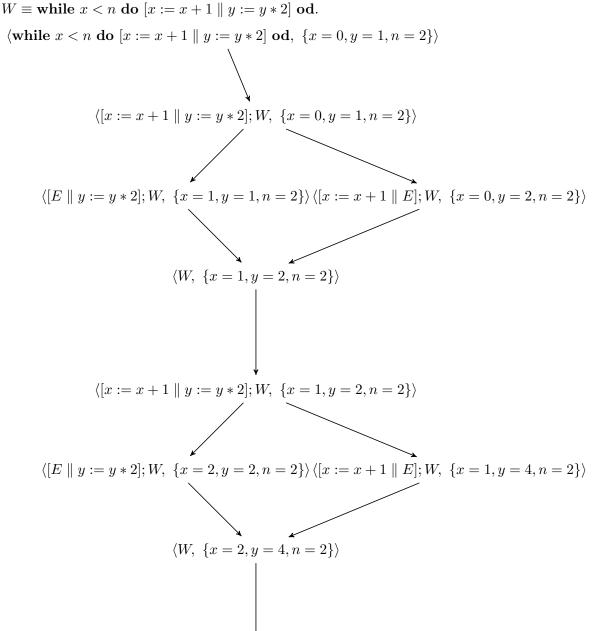
$$\mapsto k \neq b[k] \land b[k] \neq b[j]$$

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

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



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



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