CS536 Science of Programming
Fall 2024
Assignment 5 Sample Solution Sketches

1. (a) Here is the full proof outline.

$\{n > 0\}\ k := n - 1;\ \{n > 0 \wedge k = n - 1\}\ x := n;\ \{n > 0 \wedge k = n - 1 \wedge x = n\}$
$\{\textbf{inv}\ p \equiv 1 \le k \le n \wedge x = n! \div k!\}$
$\textbf{while}\ k > 1\ \textbf{do}$
$\{p \wedge k > 1\}$
$\quad \{p\ [x * k/x][k - 1/k]\}\ k := k - 1; \{p\ [x * k/x]\}\ x := x * k\ \{p\}$
$\textbf{od}$
$\{p \wedge k \le 1\}\ \{x = n!\}$

(b) Here is the minimal proof outline.

$\{n > 0\}\ k := n - 1;\ x := n;$
$\{\textbf{inv}\ p \equiv 1 \le k \le n \wedge x = n! \div k!\}$
$\textbf{while}\ k > 1\ \textbf{do}$
$\quad k := k - 1;\ x := x * k$
$\textbf{od}$
$\{x = n!\}$

2. Here is a full proof outline under partial correctness that uses backward assignments before the loop and forward assignments in the loop body.

$\{n \ge 0\}$
$\{0 \le 0 \le n \wedge 0 = sum(0, 0)\}$
$k := 0; \{0 \le k \le n \wedge 0 = sum(0, k)\} s := 0;$
$\{\textbf{inv}\ p \equiv 0 \le k \le n \wedge s = sum(0, k)\}$
$\textbf{while}\ k < n\ \textbf{do}$
$\quad \{p \wedge k < n\}$
$\quad s := s + k + 1; \{0 \le k \le n \wedge s_0 = sum(0, k) \wedge k < n \wedge s = s_0 + k + 1\}$
$\quad k := k + 1$
$\quad \{0 \le k_0 \le n \wedge s_0 = sum(0, k_0) \wedge k_0 < n \wedge s = s_0 + k_0 + 1 \wedge k = k_0 + 1\}$
$\quad \{p\}$
$\textbf{od}$
$\{p \wedge k \ge n\}\{s = sum(0, n)\}$

We need to prove the following logic implication to finish the proof:

- $n \ge 0 \Rightarrow 0 \le 0 \le n \wedge 0 = sum(0, 0)$, which is trivially true.

- $0 \leq k_0 \leq n \wedge s_0 = sum(0, k_0) \wedge k_0 < n \wedge s = s_0 + k_0 + 1 \wedge k = k_0 + 1 \Rightarrow 0 \leq k \leq n \wedge s = sum(0, k)$. Let us look at the three conjuncts on the right-hand side of the logic implication one by one. Here $k \geq 0$ is true, because $k_0 \geq 0$ and $k = k_0 + 1$. Here $k < n$ is true because $k_0 < n$ and $k = k_0 + 1$. Here $s = sum(0, k)$ is true because $s = s_0 + k_0 + 1 = sum(0, k_0) + k_0 + 1 = sum(0, k_0 + 1) = sum(0, k)$.
- $p \wedge k \geq n \Rightarrow s = sum(0, n)$. Since $p \wedge k \geq n \Leftrightarrow 0 \leq k = n \wedge s = sum(0, k)$, so $s = sum(0, n)$ is true.

3. Here is the full proof outline with forward assignment used for all assignment statements.

   $\{y \geq 1\}$ $x := 0;$ $\{y \geq 1 \wedge x = 0\}$ $r := 1;$ $\{y \geq 1 \wedge x = 0 \wedge r = 1\}$
   $\{$**inv** $p \equiv 1 \leq r = 2^x \leq y\}$
   while $2 * r \leq y$ **do**
   $\{p \wedge 2 * r \leq y\}$
   $\quad r := 2 * r;$ $\{1 \leq r_0 = 2^x \leq y \wedge 2 * r_0 \leq y \wedge r = 2 * r_0\}$ $x := x + 1$
   $\quad \{p_1 \equiv 1 \leq r_0 = 2^{x_0} \leq y \wedge 2 * r_0 \leq y \wedge r = 2 * r_0 \wedge x = x_0 + 1\}$
   $\{1 \leq r = 2^x \leq y\}$
   od
   $\{p \wedge 2 * r > y\}$ $\{r = 2^x \leq y < 2^{x+1}\}$

   We need to prove the following logic implication to finish the proof:

   - With $y \geq 1 \wedge x = 0 \wedge r = 1$, we do have loop invariant p being satisfied and we can start the loop.
   - $p_1 \Rightarrow p$. Since $1 \leq r_0 = 2^{x_0}$ and $r = 2 * r_0 \wedge x = x_0 + 1$, so $1 \leq r = 2^x$. Since $2 * r_0 \leq y$ and $r = 2 * r_0$, so $r \leq y$.
   - $p \wedge 2 * r > y \Rightarrow r = 2^x \leq y < 2^{x+1}$. Since $p \equiv 1 \leq r = 2^x \leq y$, so $r = 2^x \leq y$. Since $2 * r > y$, so $y < 2^{x+1}$.

4. Since $p \Rightarrow y \geq r$ and r increases in each iteration so $y - r$ can be used as a bound expression. (There are other bound expressions as well.)

   Here is the full proof outline under total correctness with backward assignment used for all assignment statements.

   $\{y \geq 1\}$ $\{1 \leq 1 = 2^0 \leq y\}$ $x := 0;$ $\{1 \leq 1 = 2^x \leq y\}$ $r := 1;$
   $\{$**inv** $p \equiv 1 \leq r = 2^x \leq y\}\{$**bd** $y - r\}$
   while $2 * r \leq y$ **do**
   $\{p \wedge 2 * r \leq y \wedge y - r = t_0\}$
   $\quad \{1 \leq 2 * r = 2^{x+1} \leq y \wedge y - 2 * r < t_0\}$
   $\quad r := 2 * r;$ $\{1 \leq r = 2^{x+1} \leq y \wedge y - r < t_0\}$ $x := x + 1$
   $\{1 \leq r = 2^x \leq y \wedge y - r < t_0\}$
   od
   $\{p \wedge 2 * r > y\}$ $\{r = 2^x \leq y < 2^{x+1}\}$

   We need to prove the following logic implication to finish the proof:

- $y \geq 1 \Rightarrow 1 \leq 1 = 2^0 \leq y$, which is trivially true.
- $p \wedge 2*r \leq y \Rightarrow 1 \leq 2*r = 2^{x+1} \leq y$. Since $p \equiv 1 \leq r = 2^x \leq y$, so $1 \leq 2*r = 2^{x+1}$. Since we are in the loop body, we also have $2*r \leq y$.
- $y - r = t_0 \Rightarrow y - 2*r < t_0$. This is true since $r \geq 1$.
- $p \wedge 2*r > y \Rightarrow r = 2^x \leq y < 2^{x+1}$. Proved in question 3.

5. Here is one of the possible full proof outline for this triple.

$$\begin{array}{l}
\{p\} \\
\textbf{if } sqrt(x) > y \textbf{ then} \\
\qquad \{b[x-y] = y \wedge 0 \leq (x-y) < size(b)\}\ x := b[x-y]\ \{x = y\} \\
\textbf{else} \\
\qquad \{x = b[y-x] \wedge 0 \leq (y-x) < size(b)\}\ y := b[y-x]\ \{x = y\} \\
\textbf{fi} \\
\{x = y\}
\end{array}$$

Here, precondition $p \equiv (x \geq 0) \wedge \big(sqrt(x) > y \rightarrow b[x-y] = y \wedge 0 \leq (x-y) < size(b)\big) \wedge \big(sqrt(x) \leq y \rightarrow x = b[y-x] \wedge 0 \leq (y-x) < size(b)\big)$. It is calculated using Conditional Rule 2 (total correctness version).

6. To get a safe precondition, let us start with calculating the domain of the statement.

$$\begin{aligned}
D(x := x*y; x := 1/x) &\equiv D(x := x*y) \wedge wp(x := x*y, D(x := 1/x)) \\
&\equiv T \wedge wp(x := x*y, x \neq 0) \\
&\Leftrightarrow x*y \neq 0
\end{aligned}$$

Here is a possible proof outline for this triple. We used Forward Assignment Axiom to prove both assignment statements in the program.

$$\begin{array}{l}
\{sqrt(x) \leq y \wedge x \geq 0 \wedge x*y \neq 0\} \\
x := x*y;\ \{sqrt(x_0) \leq y \wedge x_0 \geq 0 \wedge x_0*y \neq 0 \wedge x = x_0*y\} \\
x := 1/x\ \{sqrt(x_0) \leq y \wedge x_0 \geq 0 \wedge x_0*y \neq 0 \wedge x_1 = x_0*y \wedge x = 1/x_1\}
\end{array}$$

7. (a) True. Convergence is guaranteed by the existence of bound expression.

   (b) False. We need $p \Rightarrow t \geq 0$. Since p is still true after the last iteration of the loop, so $t \geq 0$ after the loop terminates.

   (c) True. Since the loop W is provable under total correctness, then by the definition of loop invariant and bound expression, we have $\vdash_{tot} \{p \wedge B \wedge t = t_0\} S \{p \wedge t < t_0\}$, which implies that $sp(p \wedge B \wedge t = t_0, S) \Rightarrow p \wedge t < t_0 \Rightarrow t < t_0$.

   (d) False. $t > 0$ does not imply that B must be true; in other words, it is possible that a loop terminates with $t > 0$.

   (e) True. It is the contra-positive of $p \Rightarrow t \geq 0$, which guaranteed by the definition of bound expression. Someone may also argue that, t should never be negative since it is a bound expression; thus $t < 0$ is false, and false implies anything is true.

8. (a) No. There is no evidence to show that $p \Rightarrow x - k + n \geq 0$.
   (b) No. There is no evidence to show that $p \Rightarrow n - k \geq 0$.
   (c) Yes. Loop invariant implies that $n - k + C \geq 0$; and k is increased after each iteration so $n - k + C$ decreases after each iteration.
   (d) No. $k - C$ increases after each iteration.
   (e) Yes. $2^n \cdot 2^{C-k} = 2^{n-k+C}$, and the power in the expression is the same as the bound expression in question 8(c). Thus, 2^{n-k+C} is non-negative and decreases after each iteration as well.

9. Here are the five possible loop invariant candidates together with the corresponding loop conditions. I will use u as the fresh variable.
   (a) $p_1 \equiv y \geq u \wedge x = 2 * y \leq n < 3 * (y + 1)$, and $B_1 \equiv u \neq 0$.
   (b) $p_2 \equiv y \geq 0 \wedge x = u * y \leq n < 3 * (y + 1)$, and $B_2 \equiv u \neq 2$.
   (c) $p_3 \equiv y \geq 0 \wedge x = 2 * y \leq u < 3 * (y + 1)$, and $B_3 \equiv u \neq n$.
   (d) $p_4 \equiv y \geq 0 \wedge x = 2 * y \leq n < u * (y + 1)$, and $B_4 \equiv u \neq 3$.
   (e) $p_5 \equiv y \geq 0 \wedge x = 2 * y \leq n < 3 * (y + u)$, and $B_5 \equiv u \neq 1$.

10. Here are the four possible loop invariant candidates together with the corresponding loop conditions.
   (a) $p_1 \equiv (z = 2^y) \wedge (2^y \leq x) \wedge (x < 2^{y+1})$, and $B_1 \equiv y < 0$.
   (b) $p_2 \equiv (y \geq 0) \wedge (2^y \leq x) \wedge (x < 2^{y+1})$, and $B_2 \equiv z \neq 2^y$.
   (c) $p_3 \equiv (y \geq 0) \wedge (z = 2^y) \wedge (x < 2^{y+1})$, and $B_3 \equiv 2^y > x$.
   (d) $p_4 \equiv (y \geq 0) \wedge (z = 2^y) \wedge (2^y \leq x)$, and $B_4 \equiv x \geq 2^{y+1}$.