Strongest Postconditions, Correctness Proofs

CS 536: Science of Programming, Spring 2018

Due Wed Mar 28, 11:59 pm

(No Late Assignments — Solution will be posted Thu Mar 29)

3/15: p.3; 3/19: pp. 2, 3; 3/24: pp.1-3; 3/29 solved

A. Instructions

- You can work together in groups of ≤ 4. Submit your work on Blackboard. Submit one copy, under the name of one person in the group (doesn't matter who). Include the names and A-IDs of everyone in the group (including the submitter) inside that copy.

B. Why?

- $sp(p, S)$ is the most information available for the result of running S when p holds.
- To prove validity of correctness triples, we use a proof system with axioms for atomic statements and rules of inference for compound statements.
- A formal proof describes the reasons for believing that something is valid.
- Proof outlines are easier to use than proofs but give the same information.

C. Outcomes

After this homework, you should be able to

- Calculate the strongest postcondition of a loop-free program.
- Compare *sp* and *wp* approaches for proving simple programs.
- Generate instances of the partial correctness proof rules.
- Verify or complete a proof of correctness for triples using basic axioms and rules of inference.
- Expand partial proof outlines to full outlines and translate full outlines to proofs.

D. Problems [100 points total]

Part 1: Strongest Postconditions [40 points] [3/24]

1. [12 = 4 * 3 points] Calculate each of the following strongest postconditions. Do only syntactic calculations, not semantic manipulations. You can use the looser sense of ≡ from lecture 14.
   a. $sp(\texttt{n} > 0, \texttt{k} := \texttt{n}; \ \texttt{r} := \texttt{k})$
   b. $sp(\texttt{k+1} \le \texttt{n} \wedge \texttt{r} = \texttt{2\^{}k}, \ \texttt{k} := \texttt{k-1}; \ \texttt{r} := \texttt{r/2})$
   c. $sp(\texttt{i} < \texttt{j} \wedge \texttt{j-i} \le \texttt{n}, \ \texttt{i} := \texttt{f(i+j)}; \ \texttt{j} := \texttt{g(i*j)})$
   d. $sp(\texttt{B(L, R)}, S)$ where $S \equiv \textbf{if } \texttt{x} < \texttt{b[M]} \textbf{ then } \texttt{R} := \texttt{M} \textbf{ else } \texttt{L} := \texttt{M} \textbf{ fi}$ and
      $\texttt{B(L, R)} \equiv 0 \le \texttt{L} < \texttt{M} < \texttt{R} < \texttt{size(b)} \wedge \texttt{b[L]} \le \texttt{x} < \texttt{b[R]}$

2. [3 points] Calculate $sp(\texttt{T}, \texttt{y} := \texttt{x}; \ \textbf{if } \texttt{x} \ge 0 \textbf{ then } \texttt{x} := \texttt{x}^2 \textbf{ else } \texttt{x} := 0 \textbf{ fi})$ and logically simplify the result. Show the result before and after simplification. If you like, you can use a `max(u, v)` function (= the larger of `u` and `v`).

3. [19 points total] Fill in the missing parts (indicated by ???) in the following rule uses.

a. [9 = 3 * 3 points]

1. {???} $x := 2*x$ {$x = 2\hat{}(k+1)$} ______ ???
2. {$x = 2\hat{}(k+1)$} ??? {$x = 2\hat{}k$} ______ ???
3. {???} $x := 2*x;\ k := k+1$ {$x = 2\hat{}k$} ______ ???

b. [5 = 2+2+1 points]

1. {$x = 2\hat{}k$} $x := x/2$ {???} assignment
2. ??? $\rightarrow x = 2\hat{}(k-1)$ predicate logic
3. {$x = 2\hat{}k$} $x := x/2$ {$x = 2\hat{}(k-1)$} ______ ???

c. [5 = 3+2 points]

1. {$p \wedge b[m] \le v$} $L := m$ {q} assumption
2. {$p \wedge b[m] > v$} $R := m$ {q} assumption
3. {p} ??? {???} conditional 1, 2

4. [6 points] Consider the triple {**inv** $x = 2\hat{}k \wedge k \le n$} **while** $k < n$ **do** $x := x*2;\ k := k+1$ **od** {p'}

a. In order to prove the triple using the loop rule, what triple must we prove first?

b. If the triple is proved using the loop rule, what must the postcondition p' be?

Part 2: Proof Rules, Proofs, and Proof Outlines [60 points] [3/24]

For the first problem, we'll be completing a proof of partial correctness of

$$\{n \ge 0\}\ k := 0;\ s := k;\ r := s;\ \{\textbf{inv}\ p\}\ \textbf{while}\ k < n\ \textbf{do}\ S_1\ \textbf{od}\ \{s = sum(n)\}$$

where $S_1 \equiv r := r+2*k+1;\ k := k+1;\ s := s+r$, the loop invariant $p \equiv 0 \le k \le n \wedge s = sum(k^2)$ [3/24] $\wedge\ r = k^2$, and $sum(k^2)$ = the sum of $0^2 \dots k^2$ (if $k \ge 0$; if $k < 0$, then $sum(k^2) = 0$). The proof below is incomplete because most of the predicates are undefined and a few rule references haven't been filled in:

1. {$n \ge 0$} $k := 0$ {p_1} Assignment
2. {p_1} $s := 0$ {p_2} Assignment
3. {$n \ge 0$} $k := 0;\ s := 0$ {p_2} [3/19] Sequence 1, 2
4. {p_2} $r := 0$ {p_3} Assignment
5. $p_3 \rightarrow p$ Predicate logic
6. {$n \ge 0$} S_0 {p_3} where $S_0 \equiv k := 0;\ s := 0;\ r := 0$ r_1 (see below) [3/19]
7. {$n \ge 0$} S_0 {p} r_2
8. {p_4} $s := s+r$ {p} Assignment
9. {p_5} $k := k+1$ {p_4} Assignment
10. {p_5} $k := k+1;\ s := s+r$ {p} Sequence 9, 8
11. {p_6} $r := r+2*k+1$ {p_5} Assignment
12. {p_6} S_1 {p} Sequence 11, 10
13. $p \wedge k < n \rightarrow p_6$ r_3
14. {$p \wedge k < n$} S_1 {p} r_4
15. {**inv** p} W {$p \wedge k \ge n$} where $W \equiv$ **while** $k < n$ **do** S_1 **od** while, 14
16. $p \wedge k \ge n \rightarrow s = sum(n)$ Predicate logic
17. {**inv** p} W {$s = sum(n)$} Postcondition weakening, 14, 15
18. {$n \ge 0$} $S_0;\ W$ {$s = sum(n)$} Sequence, 7, 17

1. [20 = 10*2 points] For parts (a) – (f), give definitions for $p_1 - p_6$. Use substitution notation [3/15] ~~and give a list of~~ but list the results of carrying out ~~of~~ the substitutions.

   (a) p_1 (b) p_2 (c) p_3 (d) p_4 (e) p_5 (f) p_6

   For parts (g) – (j), give definitions for the rule references $r_1 - r_4$. Include the line numbers (e.g. in line 3 we used Sequence 1, 2, not just Sequence).

   (g) r_1 (h) r_2 (i) r_3 (j) r_4 [3/19]

[3/24 - fix point value for Problems 2 – 4]

2. [7 points] Give the full proof outline that corresponds to lines 1 – 7 of the proof from Problem 1. (Hint: It has three statements and 5 conditions ~~four conditions~~.) Just use p_1, p_2, etc., not their expansions.

3. [5 points] Give the full proof outline that corresponds to lines 8 – 14 of the proof from Problem 1. (Hint: It begins with $p \wedge k < n$ and ends with p.) Again, use p_1, p_2, etc., not their expansions.

4. [4 points] Expand the minimal outline below into a full proof outline for partial correctness. Don't forget the implicit **else skip**.

   { T} **if** y ≥ 0 **then** x := sqrt(y) **fi** {y ≥ 0 → x = sqrt(y)}

[3/24 - fix problem number]

5. [24 = 12 * 2 points] Expand the minimal outline below into a full proof outline for partial correctness by giving definitions for $p_0 - p_7$. Also list the three predicate logic obligations. Use substitution notation in your definitions (i.e., $p_{nbr} \equiv predicate[expr/var]$), and list the results of carrying out the substitutions.

{p_0} x := 1 {p_1}; k := 0; {p_2}
{**inv** p ≡ 1 ≤ x = 2^k ≤ b[j]}
while 2*x ≤ b[j] **do**
    {p_3}
    {p_4}
    k := k+1
    {p_5}
    x := 2*x
    {p_6}
od {p_7} {q ≡ x = 2^k ≤ b[j] < 2^(k+1) }

Solution to Homework 4 — Strongest Postconditions, Correctness Proofs

Part 1 (Strongest Postconditions)

1. (Strongest postconditions)

1a. $sp(\mathtt{n > 0,\ k := n;\ r := k})$
$\equiv sp(sp(\mathtt{n > 0,\ k := n}), \mathtt{r := k})$
$\equiv sp(\mathtt{n > 0 \land k = n,\ r := k})$
$\equiv \mathtt{n > 0 \land k = n \land r = k}$

1b. $sp(\mathtt{k{+}1 \le n \land r = 2\hat{\ }k,\ \ k := k{-}1;\ r := r/2})$
$\equiv sp(sp(\mathtt{k{+}1 \le n \land r = 2\hat{\ }k,\ \ k := k{-}1}),\ \mathtt{r := r/2})$
$\equiv sp(\mathtt{k_0{+}1 \le n \land r = 2\hat{\ }k_0 \land k = k_0{-}1,\ \ r := r/2})$
$\equiv \mathtt{k_0{+}1 \le n \land r_0 = 2\hat{\ }k_0 \land k = k_0{-}1 \land r = r_0/2}$

1c. $sp(\mathtt{i < j \land j{-}i \le n,\ \ i := f(i{+}j);\ j := g(i{*}j)})$
$\equiv sp(sp(\mathtt{i < j \land j{-}i < n,\ \ i := f(i{+}j)}),\ \mathtt{j := g(i{*}j)})$
$\equiv sp(\mathtt{i_0 < j \land j{-}i_0 \le n \land i = f(i_0{+}j),\ \ j := g(i{*}j)})$
$\equiv \mathtt{i_0 < j_0 \land j_0{-}i_0 \le n \land \land i = f(i_0{+}j_0) \land j := g(i{*}j_0)}$

1d. $sp(\mathtt{B(L, R)}, S)$
$\equiv sp(\mathtt{B(L, R)},\ \mathbf{if}\ \mathtt{x < b[M]}\ \mathbf{then}\ \mathtt{R := M}\ \mathbf{else}\ \mathtt{L := M}\ \mathbf{fi})$
$\equiv sp(\mathtt{L = L_0 \land R = R_0 \land B(L, R)},\ \mathbf{if}\ \mathtt{x < b[M]}\ \mathbf{then}\ \mathtt{R := M}\ \mathbf{else}\ \mathtt{L := M}\ \mathbf{fi})$
$\equiv sp(\mathtt{L = L_0 \land R = R_0 \land B(L, R) \land x < b[M],\ \ R := M})$
$\qquad \lor sp(\mathtt{L = L_0 \land R = R_0 \land B(L, R) \land x \ge b[M],\ \ L := M})$
$\equiv (\mathtt{L = L_0 \land B(L, R_0) \land x < b[M] \land R = M}) \lor (\mathtt{R = R_0 \land B(L_0, R) \land x \ge b[M] \land L = M})$

2. $sp(\mathtt{T,\ y := x};\ \mathbf{if}\ \mathtt{x \ge 0}\ \mathbf{then}\ \mathtt{x := x^2}\ \mathbf{else}\ \mathtt{x := 0}\ \mathbf{fi})$
$\equiv sp(sp(\mathtt{T,\ y := x}),\ \mathbf{if}\ \mathtt{x \ge 0}\ \mathbf{then}\ \mathtt{x := x^2}\ \mathbf{else}\ \mathtt{x := 0}\ \mathbf{fi})$
$\equiv sp(\mathtt{y = x},\ \mathbf{if}\ \mathtt{x \ge 0}\ \mathbf{then}\ \mathtt{x := x^2}\ \mathbf{else}\ \mathtt{x := 0}\ \mathbf{fi})$
$\equiv sp(\mathtt{y = x \land x = x_0},\ \mathbf{if}\ \mathtt{x \ge 0}\ \mathbf{then}\ \mathtt{x := x^2}\ \mathbf{else}\ \mathtt{x := 0}\ \mathbf{fi})$
$\equiv sp(\mathtt{y = x \land x = x_0 \land x \ge 0,\ x := x^2}) \lor sp(\mathtt{y = x \land x = x_0 \land x < 0,\ x := 0})$
$\equiv (\mathtt{y = x_0 \land x_0 \ge 0 \land x = x_0^2}) \lor (\mathtt{y = x_0 \land x_0 < 0 \land x = 0})$

(If we continue with logical simplification)

$\Leftrightarrow \mathtt{y = x_0 \land ((x_0 \ge 0 \land x = x_0^2) \lor (x_0^2 < 0 \land x = 0))}$
$\Leftrightarrow \mathtt{y = x_0 \land x = max(x_0,\ 0)^2}$

3. (Complete formal proof)

3a.

1.	$\{\mathtt{2{*}x = 2\hat{\ }(k{+}1)}\}\ \mathtt{x := 2{*}x}\ \{\mathtt{x = 2\hat{\ }(k{+}1)}\}$	(Backward) Assignment
2.	$\{\mathtt{x = 2\hat{\ }(k{+}1)}\}\ \mathtt{k := k{+}1}\ \{\mathtt{x = 2\hat{\ }k}\}$	(Backward) Assignment
3.	$\{\mathtt{2{*}x = 2\hat{\ }(k{+}1)}\}\ \mathtt{x := 2{*}x;\ \ k := k{+}1}\ \{\mathtt{x = 2\hat{\ }k}\}$	Sequence 1,2

3b.

1.	$\{x = 2\hat{}k\}$ x := x/2 $\{x_0 = 2\hat{}k \wedge x = x_0/2\}$	(Forward) Assignment
2.	$x_0 = 2\hat{}k \wedge x = x_0/2 \rightarrow x = 2\hat{}(k-1)$	Predicate logic
3.	$\{x = 2\hat{}k\}$ x := x/2 $\{x = 2\hat{}(k-1)\}$	Postcondition weak. 1, 2

3c.

1.	$\{p \wedge b[m] \le v\}$ L := m $\{q\}$	Assumption
2.	$\{p \wedge b[m] > v\}$ R := m $\{q\}$	Assumption
3.	$\{p\}$ **if** $b[m] \le v$ **then** L := m **else** R := m **fi** $\{q\}$	Conditional 1, 2

4. (Loop rule) The loop rule says that from $\{p \wedge B\}\ S\ \{p\}$ we can prove $\{$**inv** $p\}$ **while** B **do** S **od** $\{p \wedge \neg B\}$

4a. So to use the loop rule, we need to prove

$\{x = 2\hat{}k \wedge k \le n \wedge k < n\}$ x := x*2; k := k+1 $\{x = 2\hat{}k \wedge k \le n\}$

4b. The loop's postcondition must be $x = 2\hat{}k \wedge k \le n \wedge k \ge n$

Part 2 (Correctness Proofs)

1. In the solution below, I've taken the proof and inserted the definitions of p_1, p_2, etc., the first time they are used, if they're used multiple times. If you just wrote the definitions in a list, that's fine. The original question parts have been left black.

1.	$\{n \ge 0\}$ k := 0 $\{p_1 \equiv n \ge 0 \wedge k = 0\}$	Assignment
2.	$\{p_1\}$ s := 0 $\{p_2 \equiv n \ge 0 \wedge k = 0 \wedge s = 0\}$	Assignment
3.	$\{n \ge 0\}$ k := 0; s := 0 $\{p_2\}$	Sequence 1, 2
4.	$\{p_2\}$ r := 0 $\{p_3 \equiv n \ge 0 \wedge k = 0 \wedge s = 0 \wedge r = 0\}$	Assignment
5.	$p_3 \rightarrow p$	Predicate logic
6.	$\{n \ge 0\}\ S_0\ \{p_3\}$ where $S_0 \equiv$ k := 0; s := 0; r := 0	$r_1 \equiv$ Sequence 3, 4
7.	$\{n \ge 0\}\ S_0\ \{p\}$	$r_2 \equiv$ Postcondition weakening 6, 5
8.	$\{p_4 \equiv p[s+r/s]\}$ s := s+r $\{p\}$	Assignment
9.	$\{p_5 \equiv p_4[k+1/k]\}$ k := k+1 $\{p_4\}$	Assignment
10.	$\{p_5\}$ k := k+1; s := s+r $\{p\}$	Sequence 9, 8
11.	$\{p_6 \equiv p_5[r+2*k+1/r]\}$ r := r+2*k+1 $\{p_5\}$	Assignment
12.	$\{p_6\}\ S_1\ \{p\}$	Sequence 11, 10
13.	$p \wedge k < n \rightarrow p_6$	$r_3 \equiv$ Predicate logic
14.	$\{p \wedge k < n\}\ S_1\ \{p\}$	$r_4 \equiv$ Precondition Strengthening 13, 12
	(recall $S_1 \equiv$ r := r+2*k+1; k := k+1; s := s+r)	
15.	$\{$**inv** $p\}\ W\ \{p \wedge k \ge n\}$ where $W \equiv$ **while** $k < n$ **do** S_1 **od**	while, 14
16.	$p \wedge k \ge n \rightarrow s = sum(n)$	Predicate logic
17.	$\{$**inv** $p\}\ W\ \{s = sum(n)\}$	Postcondition weakening, 14, 15
18.	$\{n \ge 0\}\ S_0;\ W\ \{s = sum(n)\}$	Sequence, 7, 17

Substitutions:

(Recall $p \equiv 0 \le k \le n \wedge s = sum(k^2) \wedge r = k^2$)

$p_4 \equiv p[s+r/s]$ $\equiv 0 \le k \le n \wedge s+r = sum(k^2) \wedge r = k^2$

$p_5 \equiv p_4[k+1/k]$ $\equiv 0 \le k+1 \le n \wedge s+r = sum((k+1)^2) \wedge r = (k+1)^2$

$p_6 \equiv p_5[r+2*k+1/r]$ $\equiv 0 \le k+1 \le n \wedge s+r+2*k+1 = sum((k+1)^2) \wedge r+2*k+1 = (k+1)^2$

2. (Full outline for lines 1 – 7 of proof)

   $\{\texttt{n} \geq 0\}$ `k := 0` $\{p_1\}$`;` `s := 0` $\{p_2\}$`;` `r := 0` $\{p_3\}\{p\}$

3. (Full outline for lines 8 – 14 of proof)

   $\{p \wedge \texttt{k} < \texttt{n}\}$ $\{p_6\}$ `r := r+2*k+1` $\{p_5\}$ `k := k+1` $\{p_4\}$ `s := s+r` $\{p\}$

   This wasn't asked for, but if you want an outline for the whole proof, we can combine the answers to Problems 2 and 3 and add in the **while** loop. (I've broken up some of the longer lines.)

   $\{\texttt{n} \geq 0\}$ `k := 0` $\{p_1\}$`;` `s := 0` $\{p_2\}$`;` `r := 0` $\{p_3\}$

   $\{$**inv** $p\}$ **while** `k < n` **do**

   $\{p \wedge \texttt{k} < \texttt{n}\}$

   $\{p_6\}$ `r := r+2*k+1`

   $\{p_5\}$ `k := k+1`

   $\{p_4\}$ `s := s+r` $\{p\}$

   od $\{p \wedge \texttt{k} \geq \texttt{n}\}$ $\{\texttt{s} = \texttt{sum(n)}\}$

4. (Expand minimal outline) Since this is the first time you're practicing this, I'll show some different possible correct answers. (There are also others, but I think these are the ones you are most likely to have found.)

   First, here's a solution that uses *wp* throughout:

   $\{\texttt{T}\}$ $\{(\texttt{y} \geq 0 \rightarrow \texttt{y} \geq 0 \rightarrow \texttt{sqrt(y)} = \texttt{sqrt(y)}) \wedge (\texttt{y} < 0 \rightarrow \texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)})\}$

   if `y` ≥ 0 **then**

   $\{\texttt{y} \geq 0 \rightarrow \texttt{sqrt(y)} = \texttt{sqrt(y)}\}$ `x := sqrt(y)` $\{\texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)}\}$

   else

   $\{\texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)}\}$ **skip** $\{\texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)}\}$

   fi $\{\texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)}\}$

   The next solution uses *sp* throughout:

   $\{\texttt{T}\}$

   if `y` ≥ 0 **then**

   $\{\texttt{y} \geq 0\}$ `x := sqrt(y)` $\{\texttt{y} \geq 0 \wedge \texttt{x} = \texttt{sqrt(y)}\}$

   else

   $\{\texttt{y} < 0\}$ **skip** $\{\texttt{y} < 0\}$

   fi $\{\texttt{y} \geq 0 \wedge \texttt{x} = \texttt{sqrt(y)} \ \vee\ \texttt{y} < 0\}$ $\{\texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)}\}$

   The third solution uses a mix of *wp* and *sp*: For the true branch, its *sp* precondition (the condition we'd use to calculate its *sp*) implies the branch's *wp*; for the false branch, its *sp* implies its *wp* postcondition.

   $\{\texttt{T}\}$

   if `y` ≥ 0 **then**

   $\{\texttt{y} \geq 0\}$ $\{\texttt{y} \geq 0 \wedge \texttt{x} = \texttt{sqrt(y)} = \texttt{sqrt(y)}\}$ `x := sqrt(y)` $\{\texttt{y} \geq 0 \wedge \texttt{x} = \texttt{sqrt(y)}\}$

   else

   $\{\texttt{y} < 0\}$ **skip** $\{\texttt{y} < 0\}$ $\{\texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)}\}$

   fi $\{\texttt{y} \geq 0 \rightarrow \texttt{x} = \texttt{sqrt(y)}\}$

5. (Expand minimal outline) Here's yet a different presentation style you might like

$\{p_0\}$ — where $p_0 \equiv \texttt{b[j]} \geq 1$

$\texttt{x} := 1\ \{p_1\};$ — $p_1 \equiv \texttt{b[j]} \geq 1 \wedge \texttt{x} = 1$

$\texttt{k} := 0;\ \{p_2\}$ — $p_2 \equiv \texttt{b[j]} \geq 1 \wedge \texttt{x} = 1 \wedge \texttt{k} = 0$

$\{\textbf{inv}\ p \equiv 1 \leq \texttt{x} = \texttt{2^k} \leq \texttt{b[j]}\}$

while $2*\texttt{x} \leq \texttt{b[j]}$ **do**

  $\{p_3\}$ — $p_3 \equiv p \wedge 2*\texttt{x} \leq \texttt{b[j]}$

  $\{p_4\}$ — $p_4 \equiv p_5[\texttt{k+1}/\texttt{k}]$

  $\texttt{k} := \texttt{k+1}$

  $\{p_5\}$ — $p_5 \equiv p[2*\texttt{x}/\texttt{x}]$

  $\texttt{x} := 2*\texttt{x}$

  $\{p_6\}$ — $p_6 \equiv p$

od $\{p_7\}$ — $p_7 \equiv p \wedge 2*\texttt{x} > \texttt{b[j]}$

$\{q \equiv \texttt{x} = \texttt{2^k} \leq \texttt{b[j]} < \texttt{2^(k+1)}\}$

Predicate Logic obligations:

$p_2 \rightarrow p$, $p_3 \rightarrow p_4$, and $p_7 \rightarrow q$.

Substitutions: Since p_4 depends on p_5, I'm listing them in reverse order (you didn't have to do that).

$p_5 \equiv p[2*\texttt{x}/\texttt{x}] \equiv 1 \leq 2*\texttt{x} = \texttt{2^k} \leq \texttt{b[j]}$

$p_4 \equiv p_5[\texttt{k+1}/\texttt{k}] \equiv 1 \leq 2*\texttt{x} = \texttt{2^(k+1)} \leq \texttt{b[j]}$