Strongest Postconditions, Proof Rules
CS 536: Science of Programming, Fall 2018
Due Wed Oct 31, 11:59 pm

A. Instructions

- You can work together in groups of \( \leq 4 \). 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.

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.
- Verify and generate instances of the partial correctness proof rules.

D. Problems [100 points total]

Part 1: Strongest Postconditions [50 points]

Properties of \( sp \) [6 points]

1. \([6 = 3+3 \text{ points}]\) Give a small example of an \( S \) such that \( \models \{ T \} S \{ sp(p, S) \} \) but \( \not\models_{\text{tot}} \{ T \} S \{ sp(p, S) \} \). (Hint: What extra information would \( \models_{\text{tot}} \{ T \} S \{ T \} \) give us?)

Calculate \( sp \) [20 points]

2. \([20 = 4 \times 5 \text{ points}]\) Calculate each of the following strongest postconditions. Do only syntactic calculations, not semantic manipulations. You can use the looser sense of \( \equiv \) from lecture.
   
   a. \( sp(n > 0, k := n; r := k) \)
   
   b. \( sp(k+1 \leq n \land r = 2 \land k, k := k-1; r := r/2) \)
   
   c. \( sp(i < j \land j-i \leq n, i := f(i+j); j := g(i+j)) \)
   
   d. \( sp(p, S) \) where \( p \equiv 0 \leq x \leq n^2 \land x = x' \) and \( S \equiv \text{if } x < n \text{ then } x := x+n \text{ else } x := x-n \text{ fi} \)

Compare \( sp \) and \( wp \) [30 points]

3. \([18 = 3 \times 6 \text{ points}]\) Calculate and logically simplify the results unless otherwise requested. (There might not be much to simplify.) Show the result before and after simplification. For the \( sp \), you're allowed to drop information about the old values of variables if you want. (But you're not required to.)
a. \( sp(x = 2^k, x := x/2) \) and \( wp(x := x/2, x = 2^k) \).

b. \( sp(T, S) \) and \( wp(S, y = \max(0, x^2)) \) where \( S \equiv \text{if } x \geq 0 \text{ then } y := x^2 \text{ else } y := 0 \text{ fi.} \)

c. \( sp(x = 2^k \land x - x = t, S) \) and \( wp(S, x = 2^k \land x - x < t) \) where \( S \equiv x := 2x; k := k+1 \).

d. \( sp(x = x_0, S) \) and \( wp(S, x = 2^k \land x - x < t) \) where \( S \equiv x := 2x; k := k+1 \).

e. \( sp(L = L_0 \land R = R_0 \land p, S) \) and \( wp(S, p) \) where \( S \equiv \text{if } x < b[L] \text{ then } R := M \text{ else } L := M \text{ fi and } p \equiv L < R \land b[L] \leq x < b[R] \). Don't simplify the \( sp \) or \( wp \).

**Part 2: Proof Rules [50 points]**

For each problem below, find a definition(s) of the predicate(s) using the proof rules. You won't always need to, but you may use predicates in later problems: \( p_1 \) is available in Problems 6 – 11, \( p_2 \) is available in 7 – 11, etc.

4. [2 points] \( p_1 \) in
   1. \( \{n \geq 0\} k := 0 \{p_1\} \) (forward) assignment

5. [2 points] \( p_2 \) in
   1. \( \{p_1\} s := 0 \{p_2\} \) (forward) assignment

6. [6 points] \( p_3 \) in:
   1. \( \{n \geq 0\} k := 0 \{p_1\} \)
   2. \( \{p_1\} s := 0 \{p_2\} \)
   3. \( \{n \geq 0\} k := 0; s := 0 \{p_3\} \) sequence 1, 2

7. [6 points] \( p_4 \) and \( p_5 \) in
   1. \( \{p_4\} k := k+1 \{p\} \) (backward) Assignment
      where \( p \equiv x = 2^k \land k \leq n \) and \( S \equiv x := x*2; k := k+1 \)
   2. \( \{p_5\} x := x*2 \{p_4\} \) (backward) assignment
   3. \( \{p_5\} x := x*2; k := k+1 \{p\} \) sequence 2, 1

8. [4 points] \( p_6 \) in
   1. \( \{p_4\} x := x*2; k := k+1 \{p\} \)
      where \( p \equiv x = 2^k \land k \leq n \) and \( S \equiv x := x*2; k := k+1 \)
   2. \( \text{inv } p \) while \( k < n \) do \( S \od \{p_6\} \) while, 1
9. [12 points] $p_7$ and $p_8$ in

1. \( \{ p_7 \} \ x := x/2 \ ; \ y := 2*y \ \{ r = X*Y-x*y \} \)
2. \( \{ p_8 \} \ x := x-1 \ ; \ r := r+y \ \{ r = X*Y-x*y \} \)
3. \( \{ r = X*Y-x*y \} \)
   
   \textbf{if} even(x) \textbf{then} x := x/2 \ ; \ r := 2*r
   
   \textbf{else} x := x-1 \ ; \ r := r+y \ \textbf{fi} \ \{ X*Y = r-x*y \} \)

\text{(Hint: Use \textit{wp.})}

10. [18 points] $p_8$, $p_9$, and $p_{10}$ in

1. \( \{ X*Y = r-x*y \land even(x) \} \ x := x/2 \ ; \ y := 2*y \ \{ p_8 \} \)
2. \( \{ X*Y = r-x*y \land \text{odd}(x) \} \ x := x-1 \ ; \ r := r+y \ \{ p_9 \} \)
3. \( \{ X*Y = r-x*y \} \)
   
   \textbf{if} even(x) \textbf{then} x := x/2 \ ; \ y := 2*y
   
   \textbf{else} x := x-1 \ ; \ r := r+y \ \textbf{fi} \ \{ p_{10} \} \)

\text{(Hint: Use \textit{sp.})}