Strongest Postconditions, Proof Rules

CS 536: Science of Programming, Fall 2019

Due Fri Nov 1, 11:59 pm

No late assignments; solution will be posted Sat Nov 2

10/24

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.

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 [50 points total]*

Lecture 13: Strongest Postconditions [21 points]

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

2. [3 points] Calculate $sp(\texttt{i} < \texttt{j} \wedge \texttt{j-i} \le \texttt{n},\ \texttt{i} := \texttt{f(i+j)};\ \texttt{j} := \texttt{g(i*j)})$. Do only syntactic calculations, not semantic manipulations. You can use the looser sense of ≡ from lecture.

3. [15 = 3 + 6 + 6 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. $sp(\texttt{x} = \texttt{2\^{}k}, \texttt{x} := \texttt{x/2})$ and $wp(\texttt{x} := \texttt{x/2}, \texttt{x} = \texttt{2\^{}k})$. (We don't get any logical simplification here.)
   b. $sp(\texttt{x} = \texttt{x}_0, S)$ and $wp(S, \texttt{odd(x)})$ where $S \equiv$ **`if`** `even(x)` **`then`** `x := x+1` **`fi`**[*]. (Don't forget the **`else skip`**.) To simplify the *sp*, assume it's okay to drop $\texttt{x}_0$ from the result.

[*] `even(x)` and `odd(x)` mean `x % 2 = 0` or `1` respectively

c. $sp(\mathtt{L} = \mathtt{L}_0 \wedge \mathtt{R} = \mathtt{R}_0 \wedge p, S)$ and $wp(S, p)$ where $S \equiv$ **if** x < b[M] **then** R := M **else** L := M **fi** and $p \equiv \mathtt{L} < \mathtt{R} \wedge \mathtt{b[L]} \le \mathtt{x} < \mathtt{b[R]}$. Don't simplify the *sp* or *wp*.

Lectures 14-15: Proof Rules [29 points]

For each problem below, find a definition(s) of the predicate(s) using the proof rules.

4. [9 = 3 * 3 points] p_1, p_2, and p_3 in in

1.	$\{p_1\}$ k := k+1 $\{p\}$	assignment
	where $p \equiv$ x = 2^k ∧ k ≤ n and $S \equiv$ x := x*2; k := k+1	
2.	$\{p_2\}$ x := x*2 $\{p_1\}$	assignment
3.	$\{p_2\}$ x := x*2; k := k+1 $\{p\}$	sequence 2, 1
4.	$p \wedge k < n \rightarrow p_2$	pred logic
5.	$\{p \wedge k < n\}$ x := x*2; k := k+1 $\{p\}$	pre str. 4, 3
6.	$\{$**inv** $p\}$ **while** k < n **do** S **od** $\{p_3\}$	while, 3

5. [8 = 2 * 4 points] q_1 and q_2 in

1.	$\{q_1\}$ x := x/2; y := 2*y {r = X*Y−x*y}	(*)
2.	$\{q_2\}$ x := x−1; r := r+y {r = X*Y−x*y}	(*)
3.	{(r = X*Y−x*y ∧ even(x) → q_1) ∧ (r = X*Y−x*y ∧ odd(x) → q_2)} **if** even(x) **then** x := x/2; r := 2*r **else** x := x−1; r := r+y **fi** {X*Y = r−x*y}	conditional 1, 2

(*) Use assignment, assignment, and sequence as in Question 4 but show just give q_1 or q_2.

6. [12 = 3 * 4 points] r_1, r_2, and r_3 in

1.	{r = X*Y−x*y ∧ even(x)} x := x/2; y := 2*y $\{r_1\}$	(*)
2.	{r = X*Y−x*y ∧ odd(x)} x := x−1; r := r+y $\{r_2\}$	(*)
3.	{r = X*Y−x*y} **if** even(x) **then** x := x/2; y := 2*y **else** x := x−1; r := r+y **fi** $\{r_3\}$	conditional 1, 2

(*) Use assignment, assignment, and sequence but just give r_1 or r_2.