Forward Assignment; Strongest Postconditions

CS 536: Science of Programming, Fall 2022

A. Why

- Sometimes, the forward version of the assignment rule is preferable to the backward version.
- The strongest postcondition of a program is the most we can say about the state a program ends in.

B. Objectives

At the end of this activity you should be able to

- Calculate the *sp* of a simple loop-free program.
- Fill in a missing postcondition of a simple loop-free program.

C. Questions

1. What basic properties does $sp(p, S)$ have?

Simple sp calculations

For Questions 2 - 7, syntactically calculate the following *sp*, showing intermediate steps. Use our extended notion of "≡" ($T \wedge p \equiv F \vee p \equiv p \wedge p \equiv p \vee p \equiv p$, basically). Simplify logically/arithmetically if you want but show the answer before and after simplification.

2. $sp(y \geq 0,$ ***skip***$)$
3. $sp(i > 0, i := i+1)$ [Hint: add an $i = i_0$ conjunct to $i > 0$]
4. $sp(k \leq n \wedge s = f(k, n), k := k+1)$
5. $sp(T, i := 0; k := i)$
6. $sp(u \leq v \wedge v - u < n, u := u+v; v := u+v)$.
7. $sp(0 \leq i < n \wedge s = sum(0, i), s := s+i+1; i := i+1)$

sp of conditionals

8. (Specify initial values late)
   a. What is $sp(x = x_0 \wedge x < 0, x := -x)$? $sp(x \geq 0,$ ***skip***$)$? What is the disjunction of these two?
   b. What is $sp(x = x_0,$ ***if*** $x < 0$ ***then*** $x := -x$ ***fi***$)$?
   c. What is the difference between your answers to parts (a) and (b)? Which of is weaker or stronger than the other?

9. Let $p \equiv x \geq y$ and $S \equiv$ ***if*** $y \geq z$ ***then*** $x := x+z$ ***else*** $y := y-z$ ***fi***.
   a. What are *rhs(S), lhs(S), free(p),* and *aged(p, S)* ?
   b. What is p_0, the version of p extended with initial value bindings?
   c. Calculate $sp(p_0, S)$. Simplify if you wish, but show the result before and after simplification.

10. Let $S \equiv$ ***if*** $x \geq 0$ ***then*** $y := x$ ***else*** $y := -x$ ***fi***.
    a. What are *rhs(S), lhs(S), free(p),* and *aged(T, S)* ?
    b. What is the significance of *aged(T, S)* being the set it is?
    c. Calculate *sp(T, S).*

© James Sasaki, 2022

Solution to Practice 13 (Forward Assignment; Strongest Postconditions)

1. The *sp* has two properties:
   - $sp(p, S)$ is a partial correctness postcondition: $\models \{p\}\ S\ \{sp(p, S)\}$.
   - $sp(p, S)$ is strongest amongst the partial correctness postconditions: $\models \{p\}\ S\ \{q\}$ iff $sp(p, S) \rightarrow q$. (Since $sp(p, S) \rightarrow sp(p, S)$, the first property is a special case of this.)

2. $y \geq 0$ (For the ***skip*** rule, the precondition and postcondition are the same.)

3. Let's implicitly add $i = i_0$ to the precondition, to name the starting value of i. Then
   $sp(i > 0, i := i+1)$
   $\equiv (i > 0)[i_0/i] \wedge i = (i+1)\ [i_0/i]$
   $\equiv i_0 > 0 \wedge i = i_0+1$

4. As in the previous problem, let's introduce a variable k_0 to name the starting value of k. Then
   $sp(k \leq n \wedge s = f(k, n), k := k+1)$
   $\equiv (k \leq n \wedge s = f(k, n))[k_0/k] \wedge k = (k+1)[k_0/k]$
   $\equiv k_0 \leq n \wedge s = f(k_0, n) \wedge k = k_0+1$

5. We don't need to introduce names for the old values of i and k (they're irrelevant).
   $sp(T, i := 0; k := i)$
   $\equiv sp(sp(T, i := 0), k := i)$
   $\Leftrightarrow sp(i = 0, k := i)$ // We've dropped the "$T \wedge$ " part of $T \wedge i = 0$)
   $\equiv i = 0 \wedge k = i$

6. Let's introduce i_0 and j_0 as we need them, then
   $sp(i \leq j \wedge j\text{-}i < n, i := i+j; j := i+j)$
   $\equiv sp(sp(i \leq j \wedge j\text{-}i < n, i := i+j), j := i+j)$
   $\equiv sp(i_0 \leq j \wedge j\text{-}i_0 < n \wedge i = i_0+j, j := i+j)$
   $\equiv i_0 \leq j_0 \wedge j_0\text{-}i_0 < n \wedge i = i_0+j_0 \wedge j = i+j_0$

7. $sp(0 \leq i < n \wedge s = sum(0, i), s := s+i+1; i := i+1)$
   $\equiv sp(sp(0 \leq i < n \wedge s = sum(0, i), s := s+i+1, i := i+1)$

   For the inner *sp*,

   $sp(0 \leq i < n \wedge s = sum(0, i), s := s+i+1)$
   $\equiv 0 \leq i < n \wedge s_0 = sum(0, i) \wedge s = s_0+i+1$ Using s_0 to name the old value of s

   Returning to the outer *sp*,

$sp(sp(0 \le i < n \wedge s = sum(0, i), s := s+i+1), i := i+1)$

$\equiv sp(0 \le i < n \wedge s_0 = sum(0, i) \wedge s = s_0+i+1, i := i+1)$

$\equiv 0 \le i_0 < n \wedge s_0 = sum(0, i_0) \wedge s = s_0+i_0+1 \wedge i = i_0+1$

8. (Specify initial values late)
   a. $x_0 < 0 \wedge x = -x_0$ and $x \ge 0$. The disjunction is $x_0 < 0 \wedge x = -x_0 \vee x \ge 0$.
   b. $x_0 < 0 \wedge x = -x_0 \vee x = x_0 \wedge x \ge 0$
   c. Part (b) is stronger because it includes $x = x_0$ in the right disjunct.

9. (sp of conditional)
   a. Let $p \equiv x \ge y \ge z$ and $IF \equiv$ ***if*** $y \ge z$ ***then*** $x := x+z$ ***else*** $y := y-z$ ***fi***.
      $lhs(IF) = \{x, y\}$, $rhs(IF) = \{x, y, z\}$, $free(p) = \{x, y, z\}$, and
      $aged(p,IF) = lhs(IF) \cap (rhs(IF) \cup free(p)) = \{x, y\} \cap (\{x, y, z\} \cup \{x, y, z\}) = \{x, y\}$.
   b. $p_0 \equiv p \wedge x = X \wedge y = Y \equiv x \ge y \wedge x = X \wedge y = Y$.
   c. First, $sp(p_0 \wedge B, S_1)$

      $\equiv sp(p_0 \wedge y \ge z,\ x := x+z)$

      $\equiv sp(x \ge y \ge z \wedge x = X \wedge y = Y \wedge y \ge z,\ x := x+z)$

      $\equiv (x \ge y \ge z \wedge y = Y \wedge y \ge z)[X/x] \wedge x = (x+z)[X/x]$

      $\equiv X \ge y \ge z \wedge y = Y \wedge y \ge z \wedge x = X+ z$.

      Then, $sp(p_0 \wedge \neg B, S_2)$

      $\equiv sp(p_0 \wedge y < z,\ y := y-z)$

      $\equiv sp(x \ge y \ge z \wedge x = X \wedge y = Y \wedge y < z,\ y := y-z)$

      $\equiv (x \ge y \ge z \wedge x = X \wedge y < z)[Y/y] \wedge y = (y-z)[Y/y]$

      $\equiv x \ge Y \ge z \wedge x = X \wedge Y < z \wedge y = Y-z$.

      So $sp(p, IF) \equiv sp(p_0 \wedge B, S_1) \vee sp(p_0 \wedge \neg B, S_2)$

      $\equiv (X \ge y \ge z \wedge y = Y \wedge y \ge z \wedge x = X+ z) \vee (x \ge Y \ge z \wedge x = X \wedge Y < z \wedge y = Y-z)$

10. (conditional sets fresh variables)
    a. If $S \equiv$ ***if*** $x \ge 0$ ***then*** $y := x$ ***else*** $y := -x$ ***fi***, $lhs(S) = \{y\}$, $rhs(S) = \{x\}$, so
       $aged(T, S) = lhs(S) \cap (rhs(S) \cup free(T)) = \{y\} \cap (\{x\} \cup \varnothing) = \varnothing$.
    b. $aged(T, S)$ being empty indicates that all the assignments in S are to fresh variables.
    c. $sp(T, S) = sp(T,$ ***if*** $x \ge 0$ ***then*** $y := x$ ***else*** $y := -x$ ***fi***)

       $\equiv sp(x \ge 0, y := x) \vee sp(x < 0, y := -x)$

       $\equiv (x \ge 0 \wedge y = x) \vee (x < 0 \wedge y = -x)$

© James Sasaki, 2022