Finding Invariants; Array Assignments

CS 536: Science of Programming, Spring 2021

Due Sun, May 9, 11:59 pm [chg 5/2]

A. Why?
• The hardest part of programming is finding good loop invariants.
• There are heuristics for finding them but no algorithms that work in all cases.
• Array assignments are harder to understand than assignments to plain variables

B. Objectives
After this homework, you should know how to
• Describe the strength connections among the conditions of \( \{ p_0 \} S_0 \{ inv \ p \} \ while \ B \ do \ S \ od \ \{ q \} \)
• Describe and use the invariant-finding heuristics "Replace a constant by a variable", “Drop a conjunct” and “Add a disjunct”.
• Array assignments aren’t like assignments to plain variables because the actual item to change can’t be determined until runtime. We can handle this by extending our notion of assignment and/or substitution.
• Perform textual substitution to replace an array element.
• Calculate the wp of an array element assignment.

C. Problems [50 points total]

Classes 19 & 20: Loop invariants 1 & 2 [26 points]
1. [10 = 5*2 points] In general, for \( \{ p_0 \} S_0 \{ inv \ p \} \ while \ B \ do \ S \ od \ \{ q \} \),
   a. Is \( p_0 \) stronger or weaker than \( p \)?
   b. Is \( p \) stronger or weaker than \( q \)?
   c. If we do zero iterations, does \( p \) still have to be true?
   d. When we start the first iteration, does \( p \) have to be true?
   e. Where inside \( S \) (if anywhere) does \( p \) have to be true?

2. [6 = 3*2 points] We’re given the postcondition \( (x^2-f(y, a) < 2*z-b) \) where \( x, y, \) and \( z \) are the variables (and are all \( \geq 0 \)), and we know \( 0 \leq a \leq n \) and \( -n \leq b \leq -1 \). If we use Replace a Constant by a Variable, then what are the candidate initialization code / invariant / loop test combinations we can get? You can ignore the power 2 as a constant to replace but not the 2 that multiplies \( z \).
3. [6 = 3*2 points] We’re given the postcondition

\[(x > 0 \lor y < n) \land (x < n \rightarrow f(x, n)) \land (f(y, n) \leftrightarrow y \geq 0)\]

If we use Drop a Conjunct, what are the candidate invariant / loop test combinations we get?

4. [4 = 2*2 points] (Add a disjunct)
   a. For the postcondition \((p_1 \land p_2)\), how are Drop a Conjunct and Add a Disjunct related?
   b. Why is Add a Disjunct less constrained than Replace a Constant by a Variable or Drop a Conjunct?

Class 21: Array Assignments [24 points]

5. [9 = 3*3 points] Syntactically calculate the following; you may simplify if you want.
   a. \(wp(b[0] := 9, \ x > b[k])\)
   b. \(wp(b[k] := b[m], \ b[m] = z)\)
   c. \(wp(b[k] := 1, \ b[k] = b[m])\)

6. [6 = 2*3] Complete the full proof outline below for partial correctness by using \(wp\) to give definitions for \(p_1\) and then \(p_2\). Logically simplify as you go.
   b. \(\{p_2\} b[x] := b[m]; \{p_1\} b[y] := b[n] \{b[x] < b[y] \land x \neq y\}\) (Hint: \(x \neq y [5/2]\) can help you simplify.)

7. [9 points] Calculate and simplify \((b[b[x]] = 8) [9 / b[y]]\). When you simplify, you can assume \(y \neq x [5/2]\). (Hint: Since you need it more than once, calculate \((b[x]) [9 / b[y]]\) first.)