Finding Invariants

CS 536: Science of Programming, Fall 2022

Due Sat Nov 12, 11:59 pm

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.

B. Objectives

After this homework, you should know how to

- Describe the strength connections among the conditions of $\{p_0\}\ S_0\ \{\textbf{\textit{inv}}\ p\}\ \textbf{\textit{while}}\ B\ \textbf{\textit{do}}\ S\ \textbf{\textit{od}}\ \{q\}$
- Describe and use the invariant-finding heuristics "Replace a constant by a variable", “Drop a conjunct” and “Add a disjunct”.

C. Problems [60 points total]

Classes 19 & 20: Loop invariants 1 & 2

1. [12 = 4*3 points] In general, for $\{p_0\}\ S_0\ \{\textbf{\textit{inv}}\ p\}\ \textbf{\textit{while}}\ B\ \textbf{\textit{do}}\ S\ \textbf{\textit{od}}\ \{q\}$,
   a. In general, roughly, is p stronger or weaker than q?
   b. When we start the first iteration, does p have to be true?
   c. Do we have to establish p if we know we'll do zero iterations?
   d. Where inside S (if anywhere) can $\neg B$ be true?

2. [9 = 3*3 points] We're given the postcondition $(x^2 - f(2*y, a) < g(z^2, b))$ where x, y, and z are the variables and $0 \le a \le n$ and $-n \le b \le -1$. Use *Replace a Constant by a Variable* to generate three different candidate invariants and loop sketches of the form

   *init. code; {**inv** invariant}* ***while*** *loop test* ***do*** *...; progress step* ***od***

   Assume $f(2*y, 0)$ and $g(z^2, -1)$ are easy to calculate. If there's no obvious way to write initialization or progress step code, say so and just give what you can. Ignore the occurrence of 2 as an exponent.

3. [9 = 3*3 points] We're given the postcondition $(x > 0 \vee y < n) \wedge (x < n \rightarrow f(x, n)) \wedge (f(y, n) \leftrightarrow y \geq 0)$. If we use *Drop a Conjunct*, what are the candidate *invariants* / *loop tests* we get? Logically simplify to get rid of as many ¬ operators as you can. You can use ⊕ for XOR if you like.

4. [4 = 2*2 points] (Add a disjunct)
   a. For the postcondition $(p_1 \wedge p_2)$, 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*?

5. [26 = 13*2 points] Take the partial proof outline for Class 20's Example 6 (Faster Multiplication), and expand it to give a full proof outline.[1] You can skip the expansion of substitutions and the listing of predicate logic obligations.[2]

[1] Yes, technically there isn't a lot of "Finding Invariants" in this problem, but it's still a good skill to practice.

[2] You need to be able to do this for the exam, but this problem is already long enough.

