Convergence; Finding Invariants; Array Assignments; Disjoint Parallelism

CS 536: Science of Programming, Fall 2018
Due Wed Nov 28, 11:59 pm [no late assignments]

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?

- To avoid runtime errors, we use domain predicates; to avoid infinite loops, we use bound functions.
- There is no algorithm for finding loop invariants, but there are some heuristics.
- We can handle array assignments if we extend our notion of substitution.
- Adding parallelism makes program execution more interesting and complicated.

C. Outcomes

After this homework, you should be able to

- State the conditions for proving loop convergence; use heuristics for finding bound functions for simple loops.
- Generate possible invariants using heuristics.
- Calculate the \( wp \) of an array element assignment using substitution for an array element.
- Recognize legal parallel programs, disjoint parallel programs, and parallel programs with disjoint conditions.

D. Problems [100 points total]

Part I: Loop Convergence [29 points]

1. \([8 = 4 \times 2 \text{ points}]\) Consider the loop \( \{ \text{inv } p \} \{ \text{bd } t \} \text{ while } j \leq n \text{ do } j := j+1 \text{ od } \). If \((p \rightarrow n \geq 0 \land 0 < C \leq j \leq n+C)\) (where \( C \) is a named constant), then for each of the following expressions, say whether or not it can be used as the bound expression \( t \) above (if not, briefly explain why).
   a. \( n \)
   b. \( n - j \)
   c. \( n + j + C \)
   d. \( n - j + 2*C \)

2. \([21 = 7 \times 3 \text{ points}]\) Expand the minimal outline below into a full proof outline for total correctness. (i.e., avoid runtime errors and loop divergence.) As part of this, define the initial precondition \( p_0 \) and initialization code \( S_0 \). The loop invariant is partially defined (extend it if you need to). Feel free to define other predicates if you find it helpful. Include a list of predicate logic obligations and show the expansion of any substitutions.

\[
\{ p_0 \} \ S_0 ; \\
\{ \text{inv } p \equiv \exists x \leq 2^k \leq b[j] \land \ ??? \} \ { \text{bd } ??? } \\
\text{while } 2*x \leq b[j] \text{ do} \\
\quad k := k+1; \\
\quad x := 2*x \\
\text{od } \{ p \land 2*x \leq b[j] \} \ { \{ x = 2^k \leq b[j] < 2^{k+1} \} } \]
Part II: Finding Invariants [25 points]

For Problems 3 and 4, let $q \equiv x \geq 0 \land z = 2^x \leq n < 2^x(x+1)$ where $n$ is constant and $x$ is a variable. Write each invariant candidate in the form $\{\text{initial precondition}\} \text{ initialization code} ; \{\text{inv invariant}\} \text{ while loop test}$. (Give an initial precondition, initialization code, etc.) You may include an educated guess about the range of the new variable, but it's not required. If you can’t find an initial precondition or initialization code for some case, explain why. Feel free to write $2^x$ as $2^x$ if you prefer.

3. [15 = 5 * 3 points] There are five possible invariants obtained by replacing a constant by a variable in $q$. Two of them involve replacing the 2 in $2^x$ and $2^x(x+1)$ and don’t lead to reasonable invariants.

(a), (b), (c) Describe the remaining three candidate invariants.

4. [9 = 3 * 3 points] (a), (b), (c) Give the candidate invariants obtained by dropping each of the three conjuncts of $q$.

Part III: Array Assignments [24 points]

5. [24 = 3 * 8 points] For each triple below, give a full proof outline for partial correctness by expanding the partial outline using wp’s. (Logically simplify as you calculate each wp.) Give definitions for the initial precondition $p$ and the intermediate condition $q$, both logically simplified.

a. $\{p\} b[j] := a; \{q\} b[i] := c \{b[j] \leq b[i]\}$

b. $\{p\} b[j] := b[m]; \{q\} b[m] := b[n] \{b[j] < b[n]\}$

c. $\{p\} b[j] := b[m]; \{q\} b[k] := b[n] \{b[j] < b[k] \land j \neq k\}$

(Hint: Logically optimize as you go with the help of $j \neq k$.)

Part IV: Parallel and Disjoint Parallelism [23 points]

6. [8 points] Let $S \equiv [x := x+5; y := x/2] || z := x/3$ and $\sigma = \{x = 12\}$. (Note that in the first thread the assignments are done sequentially.) (a) Draw an evaluation graph for $(S, \sigma)$ and (b) Give $M(S, \sigma)$.

7. [15 = 3 * 5 points] Write out a table showing, for each pair of triples, their Change, Var, and Free sets and whether the pair are parallel disjoint and/or have disjoint conditions.

- $\{x \neq y\} x := u; y := u \{x = y\}$
- $\{v = z\} z := z+1; v := v + 1 \{v = z\}$, and
- $\{w \geq u+x\} w := u+1; w := v \{w > u+x\}$