Loop Convergence & Total Correctness

CS 536: Science of Programming, Fall 2021

A. Why
• Runtime errors make our programs not work, so we want to avoid them.
• Diverging programs aren’t useful, so it’s useful to know how to show that loops terminate.

B. Objectives
At the end of this activity you should be able to
• Calculate the domain predicate of an expression.
• Show what domain predicates need to hold within a program.
• Generate possible loop bounds for a given loop.
• State the extra obligations required to prove that a partially correct program is totally correct.

C. Questions
1. Consider the triple \{inv p\} \{dec e\} while \(k < n\) do ... \(k := k+1\) od \{p ∧ k ≥ n\}. Assume \(p → n ≥ k\). To show that this loop terminates, we need a bound function \(t\) such that
   (1) \(p → n - k ≥ 0\) (which holds by assumption) and
   (2) \(p ∧ k < n ∧ t = t₀\) \(k := k+1\) \(t < t₀\). (Assume loop code before \(k := k+1\) doesn’t affect \(k\).)
   a. Can we use \(t ≡ n-k\) as a bound expression?
   b. Can we use \(t ≡ n-k+1\) as a bound expression?
   c. Can we use \(t ≡ 2n-k\) as a bound expression?
2. Use the same program as in Question 3 but assume \(p → n ≥ k-3\), not \(n ≥ k\).
   a. Why does \(n-k\) now fail as a bound expression?
   b. Give an example of a bound expression that does work.
3. Consider the loop below. (Assume \(n\) is a constant and the omitted code does not change \(k\).)
   a. Why does using just \(k\) as the bound function fail?
   b. Find an expression that involves \(k\) and prove that it’s a loop bound. (You’ll need to augment \(p\).)
      \(\{n ≥ -1\}\)
      \(k := n;\)
      \{inv p ∧ _____ \} \{dec _____ \}
while $k \geq -1$
  do ... $k := k-1$ ... od

4. What is the minimum expression (i.e., closest to zero) that can be used as a loop bound for
   
   $\{\text{inv } n \leq x+y\}$ $\{\text{dec }\}$ while $x+y > n$ do ... $y := y-1$ od ?  
   (Assume $x$ and $n$ are constant.)

5. Consider the loop $\{n > 0\}$ $k := n$; $\{\text{inv }???\}$ while $k > 1$ do ... $k := k/2$ od $\{\ldots\}$
   a. Argue that $\text{ceiling}(\log_2 k)$ is a loop bound. (Augment the invariant as necessary.)
   b. Argue that $k$ is a loop bound.
   c. Argue that $\text{ceiling}(\log_2 n)$ is not a loop bound. (Trick question.)

6. Let’s look at the general problem of convergence of $\{\text{inv } p\}$ while $B$ do $S$ od $\{q\}$. For each property below, briefly discuss whether it is (1) required, (2) allowable but not required, or (3) incompatible with the requirements.
   a. $p \rightarrow t \geq 0$
   b. $t < 0 \rightarrow \neg p$
   c. $\{p \land B \land t = t_0\}$ $S \{t = t_0 - 1\}$
   d. $p \land t \geq 0 \rightarrow B$
   e. $\neg B \rightarrow t = 0$
   f. $\{p \land B \land t = t_0\}$ $S \{t < t_0\}$

7. Argue briefly that if $s$ and $t$ are loop bounds for $W$ then so is $s+t$. (Hint: What property or properties does $s+t$ need?)
Solution to Practice 18 (Loop Termination)

1. (Termination of \{inv \ p\} \ \{dec \ n-k\} \ while \ k < n \ do \ ... \ k := k+1 \ od)
   a. Yes: \{p \land k < n \land n-k = t_0\} \ ... \ {n-(k+1) < t_0\} \ k := k+1 \ \{n-k < t_0\} \ requires
      \(n-(k+1) < n-k\), which is true.
   b. Yes: Decrementing \(k\) certainly decreases \(n-k+1\), and \(n-k+1 > n-k \geq 0\), which is the
      other requirement.
   c. Yes, but only if \(n \geq 0\): We know \(n-k \geq 0\), so \(2n-k \geq n\), which is \(\geq 0\) if \(n \geq 0\). (If \(n < 0\) then \(2n-k\) might be negative.)

2. If \(n \geq k-3\), then we only know \(n-k \geq -3\). (Note \(n-k+3\) works as a bound, however.)

3. (Decreasing loop variable)
   a. We can't just \(k\) as the bound expression because we don't know \(k \geq 0\). In fact, the
      loop terminates with \(k = -2\).
   b. Since \(k\) is initialized to \(n\), we can add \(-2 \leq k \leq n\) to the invariant and use \(k+2\) as the
      bound expression.
   c. We need to know that the invariant implies \(k+2 \geq 0\) and that the loop body de-
      creases \(k+2\).

4. The smallest loop bound is \(x+y-n\). We know it's \(\geq 0\) because \(n \leq x+y\), and we know it
   decreases by 1 each iteration, so at loop termination, \(x+y-n = 0\), which implies that
   nothing less than \(x+y-n\) can work as a bound.

5. (\(\Theta(log\ n)\) loop)
   a. Add \(0 \leq k \leq n \land n > 0\) to the invariant. Since \(k > 1\), we know \(ceiling(log_2 \ k) > 0\),
      and halving \(k\) decreases \(ceiling(log_2 \ k)\) by one and \(ceiling(log_2 \ k) - 1 \geq 0\). Thus
      \(ceiling(log_2 \ k)\) works as a loop bound.
   b. Since \(k > 1\), halving \(k\) decreases it but leaves it \(\geq 0\).
   c. \(ceiling(log_2 \ n)\) doesn't decrease because \(n\) is a constant. (Constants make terrible
      bounds :-)

6. (Loop convergence) Required are (a) \(p \rightarrow t \geq 0\), (b) \(t < 0 \rightarrow \neg p\) [i.e., \the contrapositive
   of (a)], and (f) \(\{p \land B \land t = t_0\} \ S \ \{t < t_0\}\). Property (c) \(\{p \land B \land t = t_0\} \ S \ \{t = t_0-1\}\) is
   allowable but not required: It implies (f) but is stronger than we need. Property (e)
   \(\neg B \rightarrow t = 0\) is allowable but not required. Property (d) \(p \land t \geq 0 \rightarrow B\) is incompatible with
   the requirements (it would cause an infinite loop).
7. Sum of two loop bounds. Say \( s = s_0 \) and \( t = t_0 \) at the beginning of the loop body and that \( s_0 - \Delta s \) and \( t_0 - \Delta t \) are the values of \( s \) and \( t \) at the end of the loop body. If \( s \) and \( t \) are loop bounds, then \( s > \Delta s > 0 \) and \( t > \Delta t > 0 \). For \( s + t \) to be a loop bound, we need \( 0 \leq (s_0 - \Delta s) + (t_0 - \Delta t) < s_0 + t_0 \).

Expanding, \( (s_0 - \Delta s) + (t_0 - \Delta t) = s_0 + t_0 - \Delta s + \Delta t < s_0 + t_0 \) because \( \Delta s \) and \( \Delta t \) are positive, and \( (s_0 - \Delta s) + (t_0 - \Delta t) \geq 0 \) because \( \Delta s < s_0 \) and \( \Delta t < t_0 \). So \( s + t \) is a bound function.

An interesting question you might think about: is \( s \cdot t \) a bound function?