Total Correctness: Errors and Convergence

CS 536: Science of Programming, Spring 2023

(solved) 2023-04-06: p.2, 3

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

- 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

Total Correctness

1. Complete the outline below to get a full outline for total correctness. Simplify p if it's helpful.

   {p} if x ≥ 0 then x := sqrt(x) else x:= y/b[k] fi {0 ≤ x < y}

   Hint: Use wlp to get an outline for partial correctness, then add conditions to ensure safety, then simplify.

Convergence

1. Consider the triple $\{\textbf{inv}\ p\}\ \{\textbf{bd}\ e\}\ \textbf{while}\ k < n\ \textbf{do}\ \ldots\ k := k+1\ \textbf{od}\ \{p \wedge k \geq n\}$. Assume $p \rightarrow n \geq k$. To show that this loop terminates, we need a bound function t such that

   (1) $p \rightarrow n - k \geq 0$ (which holds by assumption) and

   (2) $\{p \wedge k < n \wedge t = t_0\}\ k := k+1\ \{t < t_0\}$. (Assume loop code before $k := k+1$ doesn't affect k.)

   a. Can we use $t \equiv n - k$ as a bound expression?

   b. Can we use $t \equiv n - k+1$ as a bound expression?

   c. Can we use $t \equiv 2n - k$ as a bound expression?

2. Use the same program as in Question 3 but assume $p \rightarrow n \geq k-3$, not $n \geq k$.

   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.)

$\{n \geq -1 \wedge c \geq 0\}$
$k := n + c;$
$\{\textbf{inv}\ p \wedge \_\_\_\_\_\ \}\ \{\textbf{bd}\ \_\_\_\_\_\ \}$
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

   $\{\textbf{inv}\ n \leq x+y\}\ \{\textbf{bd}\ ...\}$ **while** $x+y > n$ **do** ... $y := y-1$ **od** ?

   (Assume x and n are constant.)

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

6. Let's look at the general problem of convergence of $\{\textbf{inv}\ p\}$ **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. [2023-04-06] Note we're not worrying about partial correctness here, just termination.
   a. $p \rightarrow t \geq 0$
   b. $t < 0 \rightarrow \neg p$
   c. $\{p \wedge B \wedge t = t_0\}\ S\ \{t = t_0 - 1\}$
   d. $p \wedge t \geq 0 \rightarrow B$
   e. $\neg B \rightarrow t = 0$
   f. $\{p \wedge B \wedge t = t_0\}\ S\ \{t < t_0\}$

7. Prove the claim that if s and t are loop bounds for W then $s+t$ is also a bound function.

Solution to Practice 18 (Loop Termination)

Total Correctness

1. First, let's use wlp to expand the outline for partial correctness. The result is

   *{p} **if** x ≥ 0 **then** {0 ≤ sqrt(x) < y} x := sqrt(x) {0 ≤ x < y}*
   __else__ {0 ≤ y/b[k] < y} x := y/b[k] {0 ≤ x < y}
   __fi__ {0 ≤ x < y}
   where $p \equiv (x \geq 0 \rightarrow sqrt(x) < y) \wedge (x < 0 \rightarrow 0 \leq y/b[k] < y)$.

   Next let's ensure that all the conditions are safe. This entails adding $x \geq 0$ for $D(sqrt(x))$, [2023-04-06] $0 \leq k < |b| \wedge b[k] \neq 0$ for $D(0 \leq y/b[k] < y)$. The result is

   *{p} **if** x ≥ 0 **then***
   {x ≥ 0 ∧ sqrt(x) < y} x := sqrt(x) {x < y}
   __else__
   {0≤ k≤|b|∧ b[k]≠ 0 ∧ 0 ≤y/b[k] < y} x := y/b[k] {0 ≤ x < y}
   __fi__ {x < y}

   where p is now $(x \geq 0 \rightarrow x \geq 0 \wedge sqrt(x) < y) \wedge (x < 0 \rightarrow 0 \leq k \leq |b| \wedge b[k] \neq 0 \wedge 0 \leq y/b[k] < y)$.

   We can simplify this to $(0 \leq k \leq |b| \wedge b[k] > 1)$.

Convergence

1. (Termination of *{__inv__ p} {__bd__ n-k} __while__ k < n __do__ … k := k+1 __od__*)
   a. Yes: $\{p \wedge k < n \wedge n-k = t_0\}$ … $\{n-(k+1) < t_0\}$ 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 ≥ 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 $n+c$, and $c \geq 0$, we can add $-2 \leq k \leq n+c$ to the invariant and use $k+2$ as the bound expression. To show it's a bound, we need (1) $p \rightarrow k+2 \geq 0$, which we can get if $p \rightarrow n \geq -1 \wedge c \geq 0 \wedge k \geq -2$ and (2) that the loop body decreases $k+2$, which it does by decrementing k.

4. The smallest loop bound is $x+y-n$. We know it's ≥ 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 $0 \leq k \leq n \wedge n > 0$ to the invariant. Since $k > 1$, we know $\text{ceiling}(\log_2 k) > 0$, and halving k decreases $\text{ceiling}(\log_2 k)$ by one and $\text{ceiling}(\log_2 k) - 1 \geq 0$. Thus $\text{ceiling}(\log_2 k)$ works as a loop bound.
   b. Since $k > 1$, halving k decreases it but leaves it ≥ 0.
   c. $\text{ceiling}(\log_2 n)$ 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) $\{p \wedge B \wedge t = t_0\}\ S\ \{t < t_0\}$. Property (c) $\{p \wedge B \wedge t = t_0\}\ S\ \{t = t_0\text{-}1\}$ 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) $p \wedge t \geq 0 \rightarrow B$ 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 Δs and Δt 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 $s*t$ a bound function?