CS 536 class 18 Thu 3/25

Loop convergence + total correctness

Program converges.

It terminates (w/o err)

Partial correctness

Termination

Convergence doesn't halt

Convergence - runtime < \infty

We won't look at

Know runtime \implies know convergence

Know # steps to termination

Don't need to know # iterations for a loop

Just need to know it's finite + decreasing

Sufficient to find lower bound on

# iterations + have that bound strictly decreasing
\[ f(n) = n^2 \]

Create a bound function/expression

Uses loop vars + program info

Attach bound func to loop keyword \texttt{and}

\[
\text{linv p} \quad \text{if \ bnd expr}\]

\text{while } B \text{ do}

\text{od}

\[ t > \# \text{ iterations left} \]

Want \( t \) to be \( \geq 0 \)

Want \( t \) decreasing

Code at \( \text{v} \) with every iteration's while test?

\[
\text{linv p} \rightarrow t \geq 0 \quad (\text{so } t \geq 0 \text{ at every while test})
\]

Decreasing \( t \) let \( t \) be a logical constant

\[
\{ p \land B \land t = t_0 = t_0^2 \} \quad \text{doing an iteration}
\]

\[ \{ p \land t < t_0^3 \} \quad \text{decreases } t \quad \text{strictly} \]
Don't need $newt = oldt + 1$

$k := n$

{ $inv \ 0 \leq k \leq n$ }
\{ $true \ k$ \}  
\{ value $k$ \}

while $k > 0$ do
  $k := k - 1$  
  value of $k$
end

$\{ k = k_0 \} \ k := k - 1 \ \{ k < k_0 \}$

$\{ k = k_0 \} \ k := k - 1 \ \{ k < k_0 \}$

$\downarrow$ downward loop $n \rightarrow n-1 \rightarrow \cdots \rightarrow 0$

$k := 0$

{ $inv \ 0 \leq k \leq n^2$ }  
$p := t > 0$

{ $true \ k \leq n \rightarrow n-k \geq 0$ }
\{ $true \ n-k \geq 0$ \}
\{ value of $n-k$ \}

while $k < n$ do
  $k := k + 1$
end

$\downarrow$ \{ $true \ n-k \geq t_0^3$ \}

$2n - k = t_0^3 \ \{ n + (k+1) \geq t_0^3 \}$  
$k := k + 1 \ \{ n+k < t_0^3 \}$
bound for can't be a constant

\[ k = n \] while \( k < n \) do \( k = k + 1 \) od

number of iterations left? \( n \) ?

can't have \( t > 0 \) \( \rightarrow \) \( B \) (loop test)

\( p \rightarrow t > 0 \) so \( p \rightarrow B \) but every while test

\( p \rightarrow t > 0 \) infinite loop

\( \vdash \) odd \( p \) tallow \( t = 0 \)

\( p / \rightarrow t > 0 \)

\( p_a \) \( \vdash t = t_0^3 \) \( \vdash \{ t < t_0^3 \} \)

decreasing \( t \)

\( t = t_0 - 1 \) not necessary.

Not required that \( t = 0 \) after loop

\( \vdash \) odd \( t = 0 \)

allowed, not

while \( k > 0 \) \( \vdash \) de... od

required

loop stops if \( t > 0 \) holds

Don't need \( k \vdash t = t_0 \)
Do need $p \cdot t = 0 \rightarrow B$

Find bound for $\frac{d}{dt} f$ as "bound"

Include at least some loop vars

loop var $x$ decreases

increasing $k$ add $x$ to candidate $t$

$-k < 0$ loop var $x$ increases subtract $x$ from

add $-x$ to candidate $t$

candidate $t < 0$ possible?

inv $k \in \mathbb{N}$ make $t$ larger add $k$ value

$n - k > 0$

Can $t > 1$ loop bound

If $t$ is a loop bound then so is $t + e$

both $f \geq 0$

If $t_1$, $t_2$ are bounds then $t_1 + t_2$ is a bound

\[
\begin{align*}
n^2 & \geq 0 \\
k_i = 0, s_i = 0
\end{align*}
\]

\[
\begin{align*}
\text{inv} & \quad 0 \leq k \leq n \\
n \cdot n = \text{sum}(0, k)
\end{align*}
\]

\[
\begin{align*}
0 & \quad \text{while } k < n \text{ do} \\
0 - k - s & \quad k := k + 1; s := s + k
\end{align*}
\]

\[
\begin{align*}
0 + n^2 - k - s & \quad \text{od} \\
n^2 + n - k - s & \quad \text{bound feasible?}
\end{align*}
\]
Binary search (roughly)

\[
\begin{align*}
&0 \leq L < R < n \land b[L..R] \text{ sorted} \\
&\text{while } R - L > 1 \text{ do} \\
&M := (R - L) / 2 \\
&\text{if } b[M] < x \text{ then } R := M \\
&\text{else } \quad L := M \\
&\end{align*}
\]

\( R := M \) makes \( R \) smaller
\( L := M \) makes \( L \) larger

bound \( 0 \\
O + R \\
O + R - L \) ?

just \( R \) ?

just \( -L \) ?

no - doesn't always decrease

\( k > L - 1 \Rightarrow R - L > 0 \) does decrease

No - \( -L < 0 \)

\( -L \) doesn't always decrease

\( \text{note doesn't decrease by 1} \)

\( \text{decreases by 1} \)

\[ \text{bound } R - L \]

\[ \text{bound } \log_b (R - L) \] is another too
To show total correctness of loop

\[ \{ p_0 \} \text{ So; } \{ \text{inv } p \} \{ \text{bd } t \} \]
while \( \beta \) do \( S \) od

1 \( \{ p_0 \} \text{ So; } \{ p \} \text{ loop initialization works} \)
2 \( \{ p \} \text{ Loop body preserves } p \)
3 \( p \rightarrow t \geq 0 \) bound \( fcn \geq 0 \)
4 \( \{ p \} \text{ Loop body } t \leq t_0 \) bound \( fcn \) decreases
5 \( p \rightarrow \beta \rightarrow q \) Loop term establishes post cond.

6 \( p \rightarrow (D(B) \land (B \rightarrow D(S))) \)

Eval of test + eval of loop body doesn't cause r/ten.

Then loop is totally correct.

\[ \{ \} \text{ p_0 \{ } \{ t \geq 0 \} \text{ while } \beta \text{ do } S \od \{ q \} \]

1, 2, 5 partial correctness.

3, 4 convergence

6 no runtime error