Chapter 7: Formalizing Programming Principles

Longest Segment Problems

Let $N \geq 0$ and let $X[0..N)$ be an integer array. Find the longest subsegment $[p..q)$ of $[0..N)$ that satisfies a certain predicate like

- all elements are zero: $(\forall i : p \leq i < q : X.i = 0)$
- the segment is left-minimal: $(\forall i : p \leq i < q : X.p \leq X.i)$
- the segment contains at most 10 zeros: $(\# i : p \leq i < q : X.i = 0) \leq 10$
- all values are different: $(\forall i, j : p \leq i < j < q : X.i \neq X.j)$

All Zeros

Determine the legngth of a longest segment of $X[0..N)$ that contains zeros only.

$$
\begin{array}{l}
|[\\
\textbf{con } N : int\ \{N \geq 0\};\ X : \textbf{array } [0..N) \textbf{ of } int; \\
\textbf{var } r : int; \\
all\ zeros \\
\{r = (\textbf{max } p, q : 0 \leq p \leq q \leq N \wedge (\forall i : p \leq i < q : X.i = 0) :\ q - p)\} \\
]|
\end{array}
$$

The post-condition is:

$$R: \; r = (\mathbf{max}\; p, q : 0 \le p \le q \le N \wedge A.p.q : \; q - p)$$

where

$$A.p.q = (\forall i : p \le i < q : X.i = 0)$$

What properties does A have?

For

$$A.p.q = (\forall i : p \le i < q : X.i = 0)$$

we have

- A holds for empty segments: $A.n.n$
- A is prefix-closed: $A.p.q \Rightarrow (\forall i : p \le i \le q : A.p.i)$
- A is postfix-closed: $A.p.q \Rightarrow (\forall i : p \le i \le q : A.i.q)$

Our invariants are from R by replacing constant N by variable n:

$$\begin{aligned} P_0 : &\quad r = (\mathbf{max}\ p, q : 0 \leq p \leq q \leq n \wedge A.p.q :\ q - p) \\ P_1 : &\quad 0 \leq n < N \end{aligned}$$

which is established by $n, r := 0, 0$.

What if $n := n + 1$?

$$
\begin{array}{ll}
 & (\mathbf{max}\ p,q : 0 \le p \le q \le n+1 \wedge A.p.q :\ q-p) \\
= & \quad \{ \text{ split off } q = n+1 \ \} \\
 & (\mathbf{max}\ p,q : 0 \le p \le q \le n \wedge A.p.q :\ q-p)\ \mathbf{max} \\
 & \qquad (\mathbf{max}\ p,q : 0 \le p \le n+1 \wedge A.p.(n+1) :\ n+1-p) \\
= & \quad \{ \ P_0 \ \} \\
 & r\ \mathbf{max}\ (\mathbf{max}\ p : 0 \le p \le n+1 \wedge A.p.(n+1) :\ n+1-p) \\
= & \quad \{ \ + \text{ distributes over } \mathbf{max} \ \} \\
 & r\ \mathbf{max}\ (n+1+(\mathbf{max}\ p : 0 \le p \le n+1 \wedge A.p.(n+1) :\ -p) \\
= & \quad \{ \text{ property of } \mathbf{max} \text{ and } \mathbf{min} \ \} \\
 & r\ \mathbf{max}\ (n+1-(\mathbf{min}\ p : 0 \le p \le n+1 \wedge A.p.(n+1) :\ p)) \\
= & \quad \{ \ \underline{\text{invariant strengthening: } Q :\ s = (\mathbf{min}\ p : 0 \le p \le n \wedge A.p.n :\ p)} \ \} \\
 & r\ \mathbf{max}\ (n+1-s)
\end{array}
$$

We thus obtain a program of the following form.

$$
\begin{array}{l}
n, r, s := 0, 0, 0; \{\text{invariant: } P_0 \wedge P_1 \wedge Q, \text{ bound: } N - n\} \\
\mathbf{do}\ n \neq N \rightarrow \\
\qquad \text{establish } Q(n := n + 1) \\
\qquad r := r\ \mathbf{max}\ (n + 1 - s); \\
\qquad n := n + 1 \\
\mathbf{od}
\end{array}
$$

How to solve the subproblem establishing $Q(n := n+1)$:

$$Q: \ s = (\mathbf{min}\ p : 0 \le p \le n \wedge A.p.n : \ p)$$

We may remove **min** to the conjunction of the following predicates:

$$\begin{array}{ll} Q_0: & 0 \le s \le n \\ Q_1: & A.s.n \\ Q_2: & (\forall p : 0 \le p < s : \neg A.p.n) \end{array}$$

Lemma. If A is prefix-closed, then

$$Q_0 \wedge Q_2 \wedge A.s.(n+1) \Rightarrow Q(n := n+1)$$

From

$$Q_0 \wedge Q_2 \wedge A.s.(n+1) \Rightarrow Q(n := n+1)$$

we can establish $Q(n := n+1)$ by considering

- $Q_0 \wedge Q_2$ as invariant,
- $\neg A.s.(n+1)$ as guard, and
- $n+1-s$ as bound function.

Theorem

$$
\begin{array}{l}
|[\\
\{A \text{ holds for empty segment and prefix-closed}\} \\
\mathbf{var}\ n, s : int; \\
n, r, s := 0, 0, 0; \\
\mathbf{do}\ n \neq N \rightarrow \\
\qquad \mathbf{do}\ \neg A.s.(n+1) \rightarrow s := s+1\ \mathbf{od}; \\
\qquad r := r\ \mathbf{max}\ (n+1-s); \\
\qquad n := n+1 \\
\mathbf{od} \\
\{r = (\mathbf{max}\ p, q : 0 \leq p \leq q \leq N \wedge A.p.q : q-p)\} \\
]|
\end{array}
$$

What is its time complexity?

Add the variable t for counting the steps.

$$
\begin{array}{l}
|[\\
\textbf{var } n, s, t : \mathit{int}; \\
n, r, s, t := 0, 0, 0, 0; \\
\textbf{do } n \neq N \rightarrow \\
\qquad \textbf{do } \neg A.s.(n+1) \rightarrow s := s+1; t := t+1 \textbf{ od}; \\
\qquad r := r \textbf{ max } (n+1-s); \\
\qquad n := n+1; t := t+1 \\
\textbf{od} \\
]|
\end{array}
$$

It is not difficult to see that $t = s + n \leq 2N$. So if $A.s.(n+1)$ can be computed in constant time, then the above program is linear.

For the all-zeros problem, how $A.s.(n+1)$ can be computed in constant time?

from

$$A.p.q = (\forall i : p \leq i < q : X.i = 0)?$$

we have

$$\neg A.s.(n+1) \;=\; (\#i : s \leq i \leq n+1 : X.i = 0) \neq 0.$$

therefore, we may introduce variable c and accompanying invariant Q' by

$$Q' : \; c = (\#i : s \leq i \leq n : X.i = 0)$$

Please complete the program now!