SMALL PROGRAMMING EXERCISES 3

M. REM

Department of Mathematics and Computing Science, Eindhoven University of Technology, 5600 MB, Eindhoven, the Netherlands

Some programming problems seem simple until one actually tries to code them. Then one easily gets lost in details and seemingly exceptional boundary cases. The 'touring robot' (Exercise 8) is an example of such a problem. Both new exercises were used in written examinations for the "Introduction to the art of programming" lectures at Eindhoven University of Technology. They turned out to be more difficult than we had expected. Exercise 7 was given in 1979 and Exercise 8 in 1976.

The specifications of the exercises contain the operator \texttt{mod}. Roughly speaking, it expresses the remainder after division. More precisely, let \( X \) and \( Y \) be integer values with \( Y \neq 0 \). Then

\[
X \div Y = q \land X \mod Y = r
\]

is equivalent to

\[
X = q \cdot Y + r \land 0 \leq r < \text{abs}(Y).
\]

In Exercise 8 a sequence of points in the plane is given. One of the potential complications of this exercise is the possibility that consecutive points are on a line parallel to the \( X \)- or \( Y \)-axis or that three consecutive points are on one line, and one of the aims of the exercise is to avoid any complications these or other ingredients may present. Both exercises allow linear solutions. None of the solutions may involve real arithmetic, i.e. all expressions must be of type integer or boolean. My solution of Exercise 7 involves the introduction of at least one auxiliary array. (For an exposition on the notation used the reader is referred to Small Programming Exercises 1, \textit{Sci. Comput. Programming} 3 (1983) 217-222.)

Exercise 7: regular polygon on a circle

\( N \) points are located on a circle and clockwise numbered \( 0, 1, \ldots, N-1 \). The arc between points \( i \) and \((i+1) \mod N \) has length \( d(i) \). The circumference of the circle is, consequently, \((\forall i: 0 \leq i < N: d(i))\). It is requested to determine whether, for given \( k \), there exist, among the points given, \( k \) points that are the vertices of a regular \( k \)-gon:

\[
[k, N: \text{int} \{3 \leq k \leq N}\}
; d(i: 0 \leq i < N): \text{array of int}
\]
\{(A: 0 \leqslant i < N: d(i) \geqslant 1) \land (S: 0 \leqslant i < N: d(i) \mod k = 0)\}

; \{b: \text{bool}\}

; S

\{b = \text{"among the } N \text{ points given there exist } k \text{ points that form a regular } k\text{-gon"}\}

\]

\]

Exercise 8: touring robot

Given are } N \text{ points, numbered } 0, 1, \ldots, N - 1, \text{ in the plane. The pair } (X(i), Y(i)), 0 \leqslant i < N, \text{ represents the Cartesian coordinates of point } i. \text{ A robot moves from point 0, via all points given, back to point 0, observing the following rules.}

(i) the robot starts in point 0 facing point 1;

(ii) the robot moves in the direction it faces;

(iii) arriving in point } i, 0 \leqslant i < N, \text{ the robot turns counterclockwise over an angle } \alpha \text{ satisfying } 0^\circ \leqslant \alpha < 360^\circ \text{ until it faces point } (i+1) \mod N;

(iv) the robot ends in point 0 facing point 1.

As a result of its tour around the points the robot has completed an integral number of turns. It is requested to determine this number:

\[[N: \text{int } \{N \geqslant 2\}]

; X, Y(i: 0 \leqslant i < N): \text{array of int}

\:{\{(A: 0 \leqslant i < N: X(i) \neq X((i+1) \mod N)}

\vee Y(i) \neq Y((i+1) \mod N))\}}

\]; \{c: \text{int}\}

; S

\{c = \text{"the number of turns of the robot"}\}

\]

\]

Solution of Exercise 3 (coincidence count)

It was requested to find a statement list } S \text{ such that

\[[M, N: \text{int } \{M \geqslant 0 \land N \geqslant 0\}]

; F(i: 0 \leqslant i < M), G(j: 0 \leqslant j < N): \text{array of int}

\{F \text{ and } G \text{ are increasing}\]
; \{ c : \textit{int} \\
; S \\
\{ c = (\text{Ni}, j : 0 \leq i < M \land 0 \leq j < N : F(i) = G(j)) \} \\
\} \\
\} \\

Define, for $0 \leq m \leq M \land 0 \leq n \leq N$,

$$C(m, n) = (\text{Ni}, j : m \leq i < M \land n \leq j < N : F(i) = G(j)).$$

The postcondition may then be formulated as

$$c = C(0, 0).$$

We introduce two variables, $m$ and $n$, and we propose to keep the following assertion invariant.

$$P : \quad c + C(m, n) = C(0, 0) \land 0 \leq m \leq M \land 0 \leq n \leq N$$

The repetition may then be initialized by "$m, n, c := 0, 0, 0$". A suitable bound function is $M - m + N - n$. Since $c(M, n) = c(m, N) = 0$, we have

$$P \land (m = M \lor n = N) \implies c = C(0, 0)$$

This leads to the choice of $m \neq M \land n \neq N$ as the guard of the repetition.

As usual, we consider the first conjunct of $P_{m+1}$, i.e.

$$c + C(m + 1, n) = C(0, 0)$$

or

$$c + C(m, n) - (\text{Nj}: n \leq j < N : F(m) = G(j)) = C(0, 0).$$

Since $G$ is increasing, we have

$$(\text{Aj}: n < j < N : G(n) < G(j))$$

and, hence,

$$(\text{Nj}: n \leq j < N : F(m) = G(j)) = \begin{cases} 
0 & \text{if } F(m) < G(n), \\
1 & \text{if } F(m) = G(n).
\end{cases}$$

We thus have

$$\{ P \land m \neq M \land n \neq N \land F(m) < G(n) \} \ m := m + 1 \{ P \}$$

and

$$\{ P \land m \neq M \land n \neq N \land F(m) = G(n) \} \ c, m := c + 1, m + 1 \{ P \}.$$
Exploiting the symmetry of the problem, we arrive at the following solution:

\[
S: \begin{align*}
& |[m, n: \text{int}] \\
& ; c, m, n := 0, 0 \{P\} \\
& ; \text{do } m \neq M \land n \neq N \\
& \rightarrow \text{if } F(m) < G(n) \rightarrow m := m + 1 \\
& \quad \Box F(m) = G(n) \rightarrow c, m := c + 1, m + 1 \\
& \quad \Box G(n) < F(m) \rightarrow n := n + 1 \\
& \quad \Box G(n) = F(m) \rightarrow c, n := c + 1, n + 1 \\
& \text{fi } \{P\} \\
& \text{vd } \{P \land (m = M \lor n = N) \text{ and, hence, } c = C(0, 0)\}
\end{align*}
\]

If so desired, the two guarded commands with equal guards may be combined into “\(F(m) = G(n) \rightarrow c, m := c + 1, m + 1, n + 1\)”.

The problem we have just tackled may be looked upon as a two-dimensional one: we were interested in the number of pairs \((i, j)\) satisfying a certain condition. This led to the introduction of two variables. We did not, however, find the invariant for these variables by just replacing in the postcondition two constants by two variables. Such an approach would have led to an invariant of the form

\[
P': c = (N, j: 0 \leq i < m \land 0 \leq j < n: F(i) = G(j))
\]

\[
\land 0 \leq m \leq M \land 0 \leq n \leq N.
\]

We would like to conclude from the conjunction of the invariant and \(m = M \land n = N\) that the postcondition holds. But \(P'\) by itself is then not sufficient. For such a conclusion to be valid, we need to extend \(P'\) with the following assertion \(Q:\)

\[
Q: \quad F(m - 1) < G(n) \land G(n - 1) < F(m).
\]

In order that \(Q\) be defined for \(0 \leq m \leq M \land 0 \leq n \leq N\), the domains of \(F\) and \(G\) must be extended in, for example, the following way:

\[
F(-1) = G(-1) = -\text{inf},
\]

\[
F(M) = G(N) = \text{inf}
\]

in which \(\text{inf}\) is the identity element of the operator \(\min\). (The identity element of \(\max\) is \(-\text{inf}\).)

As the reader may have noticed, our ‘standard invariant’ is becoming rather complicated. That is why we have chosen the approach presented earlier. The striking difference between the invariants \(P'\) and \(P\) is that in \(P'\) the computed—and
Small programming exercises

assigned to c—part of the requested answer \( C(0, 0) \) is (in form) similar to that answer, and that in \( P \) the part still to be computed—\( C(m, n) \)—is similar to the answer. I learned the solution strategy presented, which will be used in the other three solutions as well, from Jan L.A. van de Snepscheut and W.H.J. Feijen. Readers familiar with the saddleback search may notice its similarity to our solution \( S \).

Solution of Exercise 4 (dominance count)

We have to find a statement list \( S \) such that

\[
[M, N: \text{int} \{M \geq 0 \land N \geq 0\}] \\
; F(i: 0 \leq i < M), G(j: 0 \leq j < N): \text{array of int} \\
\{F \text{ and } G \text{ are ascending}\} \\
; \llbracket c: \text{int} \rrbracket \\
; S \\
\{c = (\forall i, j: 0 \leq i < M \land 0 \leq j < N: F(i) > G(j))\} \\
\rrbracket
\]

Define, for \( 0 \leq m \leq M \land 0 \leq n \leq N \),

\[
C(m, n) = (\forall i, j: m \leq i < M \land n \leq j < N: F(i) > G(j)).
\]

The postcondition, the invariant \( P \), the initialization, the bound function, and the guard are equal to those in Exercise 3. In this case the difference between \( C(m, n) \) and \( C(m + 1, n) \) is

\[
(\forall j: n \leq j < N: F(m) > G(j)).
\]

Since \( G \) is ascending, the term is 0 if \( F(m) \leq G(n) \). The problem is not symmetric in \( F \) and \( G \) and, therefore, we have to consider \( C(m, n + 1) \) as well. The difference between \( C(m, n) \) and \( C(m, n + 1) \) is

\[
(\forall i: m \leq i < M: F(i) > G(n)).
\]

Since \( F \) is ascending, the term above equals \( M - m \) if \( F(m) > G(n) \). We obtain the following solution:

\[
S: \llbracket m, n: \text{int} \rrbracket \\
; c, m, n := 0, 0, 0 \{P\} \\
; \text{do } m \neq M \land n \neq N \\
\rightarrow \text{if } F(m) \leq G(n) \rightarrow m := m + 1
\]
\( F(m) > G(n) \Rightarrow c, n := c + M - m, n + 1 \)

\[ \text{fi } \{ P \} \]

\[ \text{od } \{ P \land (m = M \lor n = N) \text{ and, hence, } c = C(0, 0) \} \]

\]

**Solution of Exercise 5 (head to tail sums)**

In this exercise we have to find a statement list \( S \) such that

\[ \{ N: \text{ int } \{ N \geq 0 \} \] ; \ F(i: 0 \leq i < N): \text{ array of int } \{ (A_i: 0 \leq i < N: F(i) > 0) \} ; \{ [c: \text{ int} \] ; \ S \}

\[ \{ c = (N_j, k: 0 \leq j < N \land 0 \leq k < N: H(j) = T(k)) \} \]

in which, for \( 0 \leq j \leq N \land 0 \leq k \leq N, \)

\[ H(j) = (S_i: 0 \leq i < j: F(i)), \]

\[ T(k) = (S_i: k \leq i < N: F(i)). \]

Notice that \( H(0) = T(N) = 0 \) and \( H(N) = T(0) \).

The exercise is very similar to the coincidence count, except that we have two functions of which one (\( H \)) is increasing and the other (\( T \)) decreasing. In the coincidence count we obtained \( C(m, n) \) from the postcondition by replacing the lower bounds of the domains of the two functions by variables. This time we replace by a variable the lower bound (0) of the domain of \( H \) and the upper bound (\( N \)) of the domain of \( T \), i.e. we define

\[ C(m, n) = (N_j, k: m \leq j < N \land 0 \leq k \leq n: H(j) = T(k)). \]

The postcondition is then, of course,

\[ c = C(0, N) \]

and the invariant

\[ P: \quad c + C(m, n) = C(0, N) \land 0 \leq m \leq N \land 0 \leq n \leq N. \]

The repetition is initialized by "\( m, n, c := 0, N, 0 \)". The bound function is \( N - m + n \).
Since
\[ c(N, n) = (Nk: 0 \leq k \leq n: H(N) = T(k)) - (Nk: 0 \leq k \leq n: T(0) = T(k)) = 1, \]
and, likewise, \( c(m, 0) = 1 \), we have
\[ P \land (m = N \lor n = 0) \Rightarrow (c + 1 = C(0, N)) \]
We, therefore, choose \( m \neq N \land n \neq 0 \) as the guard of the repetition.

By considering the first conjuncts of \( P_{m+1}^n \) and \( P_{n-1}^n \) we obtain along the same lines as in Exercise 3 a program that contains an alternative command in which \( H(m) \) and \( T(n) \) occur in the guards. In order to be able to express these guards we extend the invariant \( P \) to \( P \land Q \).

**Q:** \( h = H(m) \land t = T(n) \).

Noticing that \( H(m+1) = H(m) + F(m) \) and that \( T(n-1) = T(n) + F(n-1) \), we are led to the following solution:

**S:**
\[
\begin{align*}
\{m, n, h, t: \text{int} \\
; m, n, c := 0, N, 0 \{P\} \\
; h, t := 0, 0 \{P \land Q\} \\
; \text{do } m \neq N \land n \neq 0 \\
\rightarrow \text{if } h < t \rightarrow h, m := h + F(m), m + 1 \\
\quad \Box h = t \rightarrow c, h, m := c + 1, h + F(m), m + 1 \\
\quad \Box t < h \rightarrow t, n := t + F(n-1), n - 1 \\
\quad \Box t = h \rightarrow c, t, n := c + 1, t + F(n-1), n - 1 \\
\text{fi } \{P \land Q\} \\
\text{od } \{P \land Q \land (m = N \lor n = 0), \text{ hence, } c + 1 = C(0, N)\} \\
; c := c + 1 \{c = C(0, N)\}
\end{align*}
\]

**Solution of Exercise 6 (minimal distance)**

It is requested to find a statement list \( S \) such that
\[
\begin{align*}
\{M, N: \text{int} \} M \geq 1 \land N \geq 1 \\
; F(i: 0 \leq i < M), G(j: 0 \leq j < N): \text{array of int}
\end{align*}
\]
\{F and G are ascending\}

\(a: \text{int}\)

\(S\)

\(\{a = (\text{MIN } i, j: 0 \leq i < M \wedge 0 \leq j < N: \text{abs}(F(i) - G(j)))\}\)

By this time the reader will probably have mastered our solution strategy. With

\(C(m, n) = (\text{MIN } i, j: m \leq i < M \wedge n \leq j < N: \text{abs}(F(i) - G(j)))\)

the invariant is

\(P: \ a \min C(m, n) = C(0, 0) \wedge 0 \leq m \leq M \wedge 0 \leq n \leq N.\)

Notice that for \(m = M \vee n = N\) the formula \(C(m, n)\) concerns the \(\text{minimum}\) of the empty set. This is, by definition, \(\text{inf}\), the identity element of the operator \(\text{min}\). (The \(\text{maximum}\) of the empty set is \(-\text{inf}\).) Thus, \(C(M, n) = C(m, N) = \text{inf}\). Since

\(C(m, n) = C(m, n + 1) \min (\text{MIN } i: m \leq i < M: \text{abs}(F(i) - G(n)))\)

and since \(F\) is increasing, we have that if \(F(m) \geq G(n)\)

\(C(m, n) = C(m, n + 1) \min (F(m) - G(n)).\)

Exploiting the symmetry in \(F\) and \(G\), we obtain the following solution:

\(S: \ [[m, n, a := 0, 0, \text{inf } \{P\}]\)

\(; \ \text{do } m \neq M \wedge n \neq N\)

\(-\ \text{if } F(m) \geq G(n) \rightarrow a, \ n := a \min (F(m) - G(n)), \ n + 1\)

\(\square G(n) \geq F(m) \rightarrow a, \ m := a \min (G(n) - F(m)), \ m + 1\)

\(\text{fi } \{P\}\)

\(; \ \text{od } \{P \wedge (m = M \vee n = N), \ \text{hence, } a = a \min \text{inf } = C(0, 0)\}\)