Chapter 8

Slope Search

8.0 Introduction

Slope Search, also known as *Saddleback Search*, is a technique which is applicable to a large class of problems that involve quantifications over two bound variables, i.e., over an area contained in $\mathcal{Z} \times \mathcal{Z}$. In most applications the term of such a quantification is a monotonic function of the bound variables, for instance, ascending in both variables or increasing in one variable and decreasing in the other variable. Examples are the longest and shortest segment problems discussed in Chapter 7. For these problems the term is $q - p$, which is an increasing function of q and a decreasing function of p.

In Section 8.1 we discuss the basic principle of the slope search and we provide various examples of its use. In Section 8.2 slope search is applied to segment problems.

8.1 The basic principle

Let M and N be natural numbers and let array $f : [0..M] \times [0..N] \to \mathcal{Z}$ be ascending in both arguments, i.e.,

$$\begin{array}{l} \quad(\forall i : 0 \leq i \leq M : (\forall j : 0 \leq j < N : f.i.j \leq f.i.(j{+}1))) \\ \wedge\, (\forall j : 0 \leq j \leq N : (\forall i : 0 \leq i < M : f.i.j \leq f.(i{+}1).j)) \end{array}$$

Assume that a value X occurs in f, i.e.,

$$(\exists i, j : 0 \leq i \leq M \ \wedge\ 0 \leq j \leq N : f.i.j = X)$$

We are asked to derive a program that establishes for integer variables a and b

$$0 \leq a \leq M \ \wedge\ 0 \leq b \leq N \ \wedge\ f.a.b = X$$

Array f is ascending in both arguments. Hence, f has its minimum in $(0,0)$ and its maximum in (M,N). Since X occurs in f, we have

$$f.0.0 \le X \le f.M.N$$

Having this information, it does not help much to inspect $f.0.0$ or $f.M.N$. Two other points of $[0..M] \times [0..N]$ are possible candidates for inspection: $(0,N)$ and $(M,0)$. We consider $(0,N)$. Since f is ascending in its first argument, we have

$$f.0.N = (\mathbf{min}\, i : 0 \le i \le M : f.i.N)$$

hence,

$$f.0.N > X \;\Rightarrow\; (\forall i : 0 \le i \le M : f.i.N > X)$$

i.e., when $f.0.N > X$ then the search area may be reduced to $[0..M] \times [0..N{-}1]$. Since f is ascending in its second argument, we have

$$f.0.N = (\mathbf{max}\, j : 0 \le j \le N : f.0.j)$$

hence,

$$f.0.N < X \;\Rightarrow\; (\forall j : 0 \le j \le N : f.0.j < X)$$

i.e., when $f.0.N < X$ then the search area may be reduced to $[1..M] \times [0..N]$.

We formalize this discussion as follows. Let I and J be such that

$$0 \le I \le M \;\wedge\; 0 \le J \le N \;\wedge\; f.I.J = X$$

The 'search area' is characterized by $(I,J) \in [a..M] \times [0..b]$ or, equivalently, we choose as invariant for a repetition

$$P:\; 0 \le a \le I \;\wedge\; J \le b \le N$$

which is established by $a, b := 0, N$. The reduction of the search area in terms of P is given by the following derivations.

$$
\begin{array}{ll}
 & f.a.b < X \\
\Rightarrow & \{\, f \text{ is ascending in its second argument, } J \le b \,\} \\
 & f.a.J < X \\
\Rightarrow & \{\, f.I.J = X \,\} \\
 & a \ne I \\
\equiv & \{\, P, \text{ in particular, } a \le I \,\} \\
 & a{+}1 \le I
\end{array}
$$

and

$$
\begin{array}{ll}
 & f.a.b > X \\
\Rightarrow & \quad \{ f \text{ is ascending in its first argument, } a \le I \} \\
 & f.I.b > X \\
\Rightarrow & \quad \{ f.I.J = X \} \\
 & b \ne J \\
\equiv & \quad \{ P, \text{ in particular, } J \le b \} \\
 & J \le b{-}1
\end{array}
$$

We conclude

$$P \wedge f.a.b < X \Rightarrow P(a := a{+}1) \quad \text{and} \quad P \wedge f.a.b > X \Rightarrow P(b := b{-}1)$$

This yields the following solution:

$$
\begin{array}{l}
\quad a, b := 0, N \ \ \{\text{invariant: } P, \text{ bound: } N - a + b\} \\
;\, \mathbf{do}\ f.a.b < X \rightarrow a := a{+}1 \\
\quad [\,]\ \ f.a.b > X \rightarrow b := b{-}1 \\
\quad \mathbf{od} \\
\quad \{f.a.b = X\}
\end{array}
$$

This program has time complexity $\mathcal{O}(M{+}N)$. A similar program is obtained when we choose $(M, 0)$ as starting point.

An operational interpretation of this technique is the following. The three-dimensional surface $z = f.x.y$ has as lowest point $(0, 0, f.0.0)$ and as highest point $(M, N, f.M.N)$. Somewhere in between position X occurs. To find that position one should not start at a minimum or at a maximum, but somewhere in between, for instance, at $(0, N, f.0.N)$ or at $(M, 0, f.M.0)$, and move along the slope of the surface in such a way that position X is approximated as well as possible, i.e., by going down when the value is too high and by going up when the value is too low. Because of this interpretation, which will not be pursued any further, this technique is called Slope Search.

Note that the points where f attains its minimum or its maximum are not important. The other two points, that are either the maximum of a row and the minimum of a column, or the minimum of a row and the maximum of a column, are useful. When, for instance, f is ascending in its first argument and descending in its second argument, suitable invariants are $0 \le a \le I \wedge 0 \le b \le J$ or $I \le a \le M \wedge J \le b \le N$.

The reduction of the search area, i.e., the reduction of the problem to a smaller problem of the same form, usually leads to the introduction of a tail invariant. For the above program, we have

$$(\exists i, j : 0 \le i \le M \wedge 0 \le j \le N : f.i.j = X)$$
$$\equiv$$
$$(\exists i, j : a \le i \le M \wedge 0 \le j \le b : f.i.j = X)$$

as tail invariant. In the following sections we use tail invariants of this form.

8.1.0 Searching

In the previous section we solved the problem of searching for a value in a two-dimensional array, given that the value occurs in the array. In this section we consider the following problem: we are given integers M and N, $M \ge 0 \wedge N \ge 0$, and integer array $f[0..M)\times[0..N)$ such that f is ascending in both arguments. We are asked to determine whether value X occurs in f. A formal specification is

$$
\begin{array}{l}
[\![\, \textbf{con}\ M, N, X : \text{int}\ \{M \ge 0 \wedge N \ge 0\};\ f : \textbf{array}\ [0..M)\times[0..N)\ \textbf{of}\ \text{int}; \\
\ \{f \text{ is ascending in both arguments}\} \\
\ \textbf{var}\ r : \text{bool}; \\
\quad S \\
\ \{r \equiv (\exists i, j : 0 \le i < M \wedge 0 \le j < N : f.i.j = X)\} \\
]\!].
\end{array}
$$

Following the strategy explained in the previous section, we define 'tail' $G.a.b$ for $0 \le a \le M \wedge 0 \le b \le N$ by

$$G.a.b \equiv (\exists i, j : a \le i < M \wedge 0 \le j < b : f.i.j = X)$$

In terms of G, the post-condition of the specification may be written as

$R :\quad r \equiv G.0.N$

We introduce integers a and b and define tail invariant P_0 by

$P_0 :\quad r \vee G.a.b \equiv G.0.N$

The bounds for a and b are specified by invariant P_1 :

$P_1 :\quad 0 \le a \le M \wedge 0 \le b \le N$

A proper initialization of $P_0 \wedge P_1$ is $a, b, r := 0, N, \text{false}$. For $a = M \vee b = 0$ the range of the quantification in G is empty, hence,