MPL Decision Structures

Decision structures provide a way to control the flow in an algorithm. MPL provides three decision structures. The simplest one is the **if** structure which has the general form shown in Figure 1-(a). The expression *condition* is a logical (or relational) expression that evaluates to one of the logical constants **true** or **false**. Each T_i is either a mathematical expression, an assignment statement, another decision structure, or an iteration structure (described below).

The **if** structure usually operates in the following way: when *condition* evaluates to **true**, the indented[1] statements $T_1, T_2, \ldots, T_m$ are evaluated, and when *condition* evaluates to **false** these statements are skipped. The exception to this scheme arises when the **if** statement is included in a procedure, and one of the indented statements includes a *Return*. In this case, when *condition* is **true**, the statements controlled by the **if** are evaluated until the *Return* is encountered, at which point the procedure terminates, and the evaluated form of the argument to *Return* is returned by the procedure. This exception also applies to the other decision and iteration structures described below.

A more general decision structure is the **if-else** structure which allows for two alternatives. It has the general form[2] shown in Figure 1-(b). When the expression *condition* evaluates to **true**, the statements $T_1, \ldots, T_m$ are evaluated, and when *condition* evaluates to **false**, the statements $F_1, \ldots, F_n$ are evaluated.

Example 1 Here is a simple example of an **if-else** structure:

$$\begin{aligned} &\textbf{if}\ \ 0 \le x \ \textbf{and}\ x \le 1\ \textbf{then} \\ &\quad f := x^2 + 4 \\ &\textbf{else} \\ &\quad f := x^2 - 1; \end{aligned} \tag{1}$$

The most general MPL decision structure is the *multi-branch* decision structure which allows for a sequence of conditions. It has the general form shown in Figure 2. In this generality, the structure contains zero or

[1] Some computer algebra languages require a termination symbol (such as `end_if`, `fi`, or `]`) to indicate the extent of statements controlled by the **if** structure. In MPL, these statements are indicated by indentation without a termination symbol.

[2] As is common practice in some programming languages, in MPL we omit the semicolon at the end of a statement that precedes an **else**, an **elseif** (defined below), and an **End**.

if *condition* **then**
  T_1;
  T_2;
  $\vdots$
  T_m;

(a) The **if** structure.

if *condition* **then**
  T_1;
  T_2;
  $\vdots$
  T_m
else
  F_1;
  F_2;
  $\vdots$
  F_n;

(b) The **if-else** structure.

Figure 1. The general form of the MPL **if** and **if-else** decision structures.

if $condition_1$ **then**
  S_{11};
  S_{12};
  $\vdots$
  S_{1m_1}
elseif $condition_2$ **then**
  S_{21};
  S_{22};
  $\vdots$
  S_{2m_2}

  $\vdots$

elseif $condition_n$ **then**
  S_{n1};
  S_{n2};
  $\vdots$
  S_{nm_n}
else
  F_1;
  F_2;
  $\vdots$
  F_r;

Figure 2. The MPL multi-branch structure that provides for a sequence of alternatives.

more **elseif** sections and an optional **else** section. Upon evaluation, the logical expressions $condition_1, condition_2, \ldots$ are evaluated in sequence. If $condition_i$ is the first one that evaluates to **true**, then the statements in that section $S_{i1}, \ldots, S_{im_i}$ are evaluated while all the other statements are skipped. If none of the tests evaluate to **true**, the statements in the **else** section (if included) are evaluated.

An example that uses **elseif** blocks is the *Automatic_Simplify* procedure in Figure 3.10 (page 92). (Implementation: Maple (mws), Mathematica (nb), MuPAD (mnb).)

All computer algebra languages provide decision structures (Figure 3).

MPL	Maple	Mathematica	MuPAD
if	`if`	`If`	`if`
if-else	`if-else`	`If`	`if-else`
if-elseif-else (multi-branch)	`if-elseif-else`	`Which`	`if-elseif-else`

Figure 3. MPL decision structures and the corresponding structures in Maple, Mathematica, and MuPAD. (Implementation: Maple (mws), Mathematica (nb), MuPAD (mnb).)

Return to Chapter 1, page 3