# INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

## ATLAS COMPUTING SERVICE

ALGOL BULLETIN NUMBER 1 19th March, 1964

The first version of the Atlas Algol compiler is working and available for use by the computing service; this is the first of a series of occasional bulletins containing information concerning diagnostic facilities and improvements and changes to the compiler.

## 1. Code procedures

The compiler will accept a procedure whose body is written in a form of machine code.

### 1.1 Syntax

The phrases defining the permissible form of a code body are as follows, using the notation of the compiler compiler.

```
PHRASE [code] = begin code [cl*?] [instruction ][; instruction] end

PHRASE [; instruction] = [; cl] [instruction] [; instruction], NIL

PHRASE [;cl] = ; [cl*?], [ERROR]

PHRASE [cl] = [N]:

PHRASE [instruction]=[ FD] [,][ N][ ,] [Bm and addr],

                      [add 1]/[cl*?] [add 1],

                      [±'] [K],

                       B79 = ADDRESS OF [ID] [ [ ] [ax] [,ax]],

                      [ ERROR],

                       NIL

PHRASE[ Bm and addr]= - [,][ID],

                      [N] [,] [add 1]

PHRASE [add 1] = [±'] [ add pr] [ ± add pr ]

PHRASE [add pr] = [N]. [OD], [N] , *[OD*] , ([N])

PHRASE [± add pr] =[ ±][ add pr] [ ± add pr], NIL

PHRASE [, ax] = [ , ] [ax] [ , ax], NIL

PHRASE [±'] = +, -, NIL

PHRASE [±] = +, -
```

The phrases whose meanings are assumed in the above definitions are:-

[N] representing the class of integer constants,

[K] representing the class of Algol numbers as defined in paragraph 2.5.1 of the revised report,

[ID] representing the class of Algol identifiers,

[ax] representing the class of Algol arithmetic expressions,

[FD] representing the class of Atlas function digits,

[OD] representing the class of octal digits

([ERROR] represents the error procedure within the compiler which is entered in the event of failure to recognise a valid syntactical structure.)

### 1.2 Semantics and restrictions

A code body consists of a sequence of instructions and other items, separated by semicolons and enclosed between the symbols begin code and end. (As in other parts of Algol, layout features such as spaces and newline are ignored.) An [instruction] may be labelled by an integer in the range 0-127 followed by a colon, as may the second half word of the second category of [instruction]. Each [instruction] occupies a full Atlas word and the sequence is entered immediately following the symbol code and left through the symbol end.

Reference may be made by means of their identifiers, to the formal parameters of the procedure, to the name of the procedure if it is a type procedure and to variables and labels global to the procedure.

To facilitate references to arrays, the form of instruction

B79 = ADDRESS OF [ID] [[ ] [ax] [,ax]]

is provided, giving in B79 the address of an individual subscripted variable. For example,

B79 = ADDRESS OF alpha [ 2,1 ]

It is expected that array elements will shortly be stored consecutively "by rows"; this property should not be used until confirmed in a later bulletin.

The octal digit representation of a half word is limited to at most 8 digits; if there are less than 8 digits these are understood to be left-adjusted. For example *543 is equivalent to *54300000.

### 1.3 Example

The following is the procedure declaration for the procedure sign, which forms part of the permanent material.

```
integer procedure sign (E);

value E; real E;

begin code 324, 0, -, E;
           234, 127, 0, (1);
           237, 127, 0, (2);
           324, 0, 0, (3);
           121, 127, 0, (1);
        3: + 1;
        2: 325, 0, 0, (3);
        1: 356, 0, -, sign

end
```

## 2. Faults found during execution

This section gives information concerning the diagnostic printing which is provided following a fault during program execution. The fault numbers described in section 4.2 of CS.378 (Provisional Reference Manual) are superseded.

Briefly, information is provided in the following order:-

(a) Error message

(b) Contents of B-lines

(c) Contents of incomplete output buffer

(d) An 'error trace' of the sequence of activations of blocks and procedures backwards from the point of error. This trace refers to blocks and procedures by means of the serial numbers allocated by the compiler and printed as part of the monitor outline of the program (see section 4.1 of CS.378) and consists of firstly, the number of the block or procedure in which the error occurred, followed by the number of the block or procedure from which that was called or activated, and so on back to the outermost level of the program.

Each block number is accompanied by a list of current values of certain of the real and integer variables declared in the block head, and each procedure number by a list of values of the actual parameters at the time of call.

This brief information is now considered in more detail.

### 2.1 Error messages and contents of B-lines

For faults detected by hardware, extracodes or supervisor, the message is the standard supervisor message, accompanied by a print of the three Atlas instructions in the program at the point of failure. The contents of the B-lines are at present printed because of their possible interest to the compiler writers; eventually this printing may be suppressed. Four categories of fault are detected by the compiled program: printed messages are:-

```
ILLEGAL COMPOSITE CHAR
ARRAY UB < LB
DATA FAULT.   CHAR =
     followed by the offending character, if
     this is available on the printer.
ILLEGAL ASSIGNMENT
```

## 2.2 Contents of incomplete output buffer

Output is handled by the compiled program through a one-line buffer, the contents of which are passed to the output stream when a newline character is called for. Consequently, following a supervisor fault, an incomplete line of output will in general remain to be printed after the contents of the B-lines.

## 2.3 Current values of variables and parameters

Following each block number in the 'error trace' the current values of the last set of integers and the last set of real quantities declared in the block head are output, in that order. (Values of array elements are not output.) Integer values, if any, are printed eight per line in an integer format and real values, if any, are printed seven per line in a standard floating point format.

It should be noted that only the last declarations of each type in the block head are considered. Thus if an error occurs in a block whose heading is begin real a; integer k; real b, c; integer l, m, n; the values of l, m, n will be printed on one line, followed by the values of b, c on another, and values of k and a will not appear.

If a variable has no defined value at the time of the error, the value printed is of no significance.

Following each procedure number in the 'error trace', the contents of information words, one of which corresponds to each actual papameter, are printed as if they were real numbers. The values are significant only in the case of expressions substituted for formal input parameters called by value. If a parameter is of type Boolean then false will be represented by a zero and true by a non-zero number.

In the case of standard machine errors, when it is possible for a few instructions beyond the faulty one to have been obeyed before the program is halted, there is a chance that the block in which the error occurred will occasionally be left. In this case, information relating to the block is lost.

## 2.4 Standard procedure numbers

The standard functions and input/output procedures are declared in a block of permanent material enclosing the user's program. The serial numbers currently assigned to these are listed in the following table. The list may be extended from time to time.

| Procedure name | Serial number |
|---|---|
| abs | 1 |
| sign | 2 |
| sqrt | 3 |
| sin | 4 |
| cos | 5 |
| arctan | 6 |
| ln | 7 |
| exp | 8 |
| entier | 9 |
| select input | 10 |
| select output | 11 |
| read | 12 |
| read Boolean | 13 |
| output | 14 |
| print | 15 |
| write text | 16 |
| space | 17 |
| newline | 18 |
| paper throw | 19 |
| write Boolean | 20 |

## 3. Specification of the procedure output

The procedure 'output' incorporated in the permanent material is an addition to the list given in CS.395 (Provisional specification of input/output procedures). Its procedure heading is as follows:-

```
procedure output (quantity); value quantity;
                    real quantity;

          comment  Outputs to the currently selected output
          document the value of quantity in a standard floating-
          point form, followed by a semicolon and a newline
          character.  The layout is that produced by output of
          the number by a call of procedure 'print' with
          m = 0, n = 10.
```

## 4. A note on arrays

The efficiency of a compiled program handling array elements is enormously increased when the subscript expressions consist either of integer variables, integer constants or combinations of these by addition and subtraction only.