

I. ALGOL-M LANGUAGE DESCRIPTION

A. FEATURES OF THE ALGOL-M LANGUAGE

Although ALGOL-M was modeled after ALGOL-60, no attempt was made to make it a formal subset of ALGOL-60. This was done intentionally in order to provide a language which would be best suited to the needs of applications programmers using microcomputer systems. However, the basic structure of ALGOL-M is similar enough to ALGOL-60 to allow simple conversion of programs from one language to the other. This was considered particularly important in view of the fact that the standard publication language is ALGOL-60. Therefore, there exists a large source of applications programs and library procedures which can be simply converted to execute under ALGOL-M.

1. Type Declarations

ALGOL-M supports three types of variables: integers, decimals, and strings. Integers may be any value between -16,383 and +16,383. Decimals may be declared with up to 18 digits of precision and strings may be declared as long as 255 characters. The default precision for decimals is ten digits and the default length for strings is ten characters. Decimal and string variable lengths may be integer variables which can be assigned actual values at run-time.

Another form of declaration in ALGOL-M is the array

declaration. Arrays may have up to 255 dimensions with each dimension ranging from 0 to +16,383. The maximum 8080 microprocessor address space of 63k bytes limits practical array sizes to something smaller than the maximum. Dimension bounds may be integer variables with the actual values assigned at run-time. Arrays may be of type integer, decimal or string.

2. Arithmetic Processing

Integer and binary coded decimal arithmetic are supported under ALGOL-M. Integers may be used in decimal expressions and will be converted to decimals at run-time. The integer and decimal comparisons of less-than (<), greater-than (>), equal-to (=), not-equal-to (<>), less-than-or-equal-to (<=), and greater-than-or-equal-to (>=) are provided. Additionally, the logical operators AND, OR and NOT are available.

3. Control Structures

ALGOL-M control structures consist of BEGIN, END, FOR, IF THEN, IF THEN ELSE, WHILE, CASE and GOTO constructs. Function and procedure calls are also used as control structures. ALGOL-M is a block structured language with a block normally bracketed by a BEGIN and an END. Blocks may be nested within other blocks to nine levels. Variables which are declared within a block can only be referenced within that block or a block nested within that block. Once program control proceeds outside of a block in which a variable has been declared, the variable may not be

referenced and, in fact, run-time storage space for that variable no longer exists.

Functions, when called, return an integer, decimal

or string value depending on the type of the function. Procedures do not return a value when called. Both functions and procedures may have zero or more parameters which are call by value and both may be called recursively.

3. Input/Output

The ALGOL-M WRITE statement causes output to the console on a new line. The desired output is specified in a write list which is enclosed in parentheses. String constants may be used in a write list and are characterized by being enclosed in quotation marks. Any combination of integer, decimal and string variables or expressions may also be used in a write list. A WRITEON statement is also available which is essentially the same as the WRITE statement except that output continues on the same line as the output from a previous WRITE or WRITEON statement. When a total of 80 characters have been written to the console, a new line is started automatically. A TAB option may also be used in the write list which causes the following item in the write list to be spaced to the right by a specified amount.

Console input is accomplished by the READ statement followed by a read list of any combination of integer, decimal and string variables enclosed in parentheses. If embedded blanks are desired in the input for a string

variable, the console input must be enclosed in quotation marks. A READ statement will result in a halt in program execution at run-time until the input values are typed at the console and a carriage return is sent. If the values typed at the console match the read list in number and type,

ALGINTRO.TXT

program execution continues. If an error as to number or type of variables from the console occurs, program execution is again halted until values are re-entered on the console.

5. Disk Access

ALGOL-M programs may read data from, or write data to, one or more disk files which may be located on one or more disk drives. When file input or output is desired, the appropriate READ or WRITE statement is modified by placing a filename identifier immediately after READ or WRITE. The actual name of the file may be assigned to the file name identifier when the program is written or it may be assigned at run-time. Various disk drives are referenced by the letters A through Z. A specific drive may be specified by prefixing the actual file name with the desired drive letter followed by a colon. Additionally, if random file access is desired, the file name identifier may be followed by a comma and an integer constant or variable. This integer value specifies the record within the file which is to be used for input/output.

Prior to the use of a file name identifier in a READ or WRITE statement, the file name identifier must appear in a file declaration statement. The file name identifier can

4

only be referenced within the same block (or a lower block) as the file declaration. Files are normally treated as unblocked sequential files. However, if blocked files are desired, the record length may optionally be specified in brackets after the file name identifier in the file declaration statement.