



## A BRIEF DESCRIPTION OF JOVIAL

T. E. Cheatham, Jr.  
Harvard University  
Aiken Computation Laboratory  
Cambridge, MA 02138

JOVIAL (Jules' Own Version of the International Algebraic Language) was one of the first programming languages developed primarily to aid in programming large complex real time systems. Today it remains a major language for these applications and versions of JOVIAL have been implemented on dozens of different computers. The language and its compilers have been developed primarily by the Systems Development Corporation.

JOVIAL was based on Algol-58 but includes numerous features not in Algol (58 or 60) which make it particularly useful for programming large scale systems. The most important of these is the COMPOOL (communications POOL), a central repository of data descriptions which permits programmers to reference data items without concerns as to how they are represented on some particular computer.

The structure of JOVIAL programs is very similar to those of ALGOL. JOVIAL provides a block structure for compound statements; its statement structure, loop structures, and procedure calls are very similar to those of ALGOL.

Basic data items include integer and real numbers, booleans, strings, and status items for example, RED, BLUE, GREEN may be the three possible values of some status items. If the variable  $x$  were such an item then the assignment

$X = \text{RED}$

and the equality relation

$X \text{ EQ } \text{GREEN}$

are permitted.

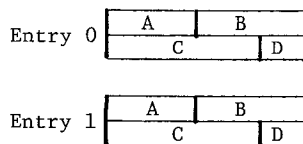
There are the usual arithmetic, relational, and logical operators and it has a conditional statement which can have multiple arms. Thus one can write

```
IFEITH P $ I = 1 $
ORIF  Q $ I = 2 $
ORIF  R $ I = 3 $
END
```

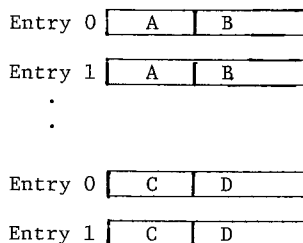
To set  $I$  to 1, 2, or 3 or leave it unmodified

according to the truth of the booleans  $P$ ,  $Q$ , and  $R$ .

JOVIAL differs markedly from ALGOL (and most other languages) in the area of data definitions. At the lowest level, JOVIAL provides for items (basic data elements like numbers, strings, and so on). At the next level are entries - ordered collections of items (they would be called structure or records today). At the top level are tables, which are sequences of entries. JOVIAL has means for the user to control how tables are actually represented in some computer, permitting for example, the user to call for "serial" or "parallel" storage of tables to simplify indexing operations when more than one word of storage is required to accommodate all items in an entry. In serial tables all words containing an entry are contiguous;



A parallel table with the same items is depicted



JOVIAL also permits machine language code inserts and provides some pseudo operations which reference the COMPOOL to provide the shifting and masking operations required to extract items from an entry thus relieving programmers from being concerned with the detailed structure of tables and permitting this structure to change without programs being adversely affected. For example the code sequence

```
CLA ITEM      Move ITEM to accumulator
ETR ITEM      AND with a mask.
```

POS ITEM      Position least significant bit  
                 to be rightmost

will extract ITEM from the word containing it and  
position it in the accumulator.