The Multistore system was developed in order to recog-
nize and explain structural patterns in natural-language
sentences (specifically English) and eventually yield an
output in which the relations between the various items of
the sentence are hierarchically displayed.

The recognition of these structural patterns is made by
means of a system of rules which operate on a sequence of
words, i.e. a sentence, whose individual characteristics
are pre-established. By individual characteristics are meant
the possibilities a word has to correlate (i.e. to form a
syntactic combination) with another item; these possi-
bilities are represented by 'correlators', that is, by
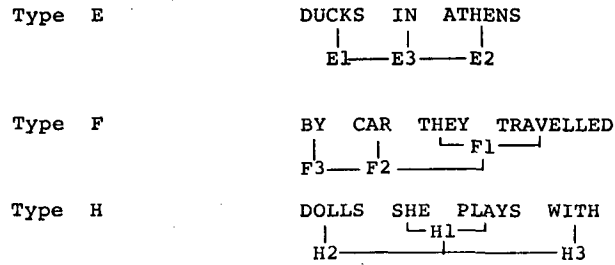syntactic elements which link two items in a correlation.

Each word is characterized by a set of pre-established
data:

a) the S-code, which distinguishes between the various
   senses of a homograph. For instance, a word like "READ"
   will have four different S's to distinguish between:

READ = supine	e.g.	I CAN READ
READ = past tense		YESTERDAY I READ
READ = past part.		I HAVE READ
READ = noun		A LONG READ

   These distinctions are essential, since whenever a homo-
   graph occurs, one and only one of its meanings can be
   taken into consideration to make the final pattern, un-
   less, of course, the sentence is ambiguous and more than

one final pattern is to be recognized, as in:

```
                        i) present tense
        I READ THE BOOK
                        ii) past tense
```

b) the sequence of correlational indices (Ic's), that is,

the string of potential links that each word-sense has.

Each Ic represents a possible syntactic connection be-

tween two items and is identified by:

1) the code number of the relation it establishes between

two items;

2) the 'type' of correlation. There are six different types

of correlation which split into two groups:'explicit'

correlators and 'implicit' correlators.

By 'explicit' correlator we mean a linking element which is

represented by a linguistic item; prepositions and conjunct-

ions are explicit correlators; by 'implicit' correlator we

mean a relation between two items, which is not expressed

by any linguistic item but is indicated by the relative po-

sition of the two items (which we call their correlational

function).

```
          IMPLICIT  CORRELATOR

    Type  N                  I      AM
                             |      |
                             N1————N2

    Type  M                  AM     I
                             |      |
                             M2————M1

    Type  V                  SERIOUSLY, HE LEFT
                                  |      └V2┘
                                  V1————————┘
```

EXPLICIT CORRELATOR

Type E   DUCKS   IN   ATHENS
                             | | |
                            E1——E3——E2

Type F   BY   CAR   THEY   TRAVELLED
                            | | └— F1——┘
                           F3—— F2 ————————┘

Type H   DOLLS   SHE   PLAYS   WITH
                             | └—H1—┘ |
                            H2——————————┴————————H3

For each type there are different correlational func-
tions which determine the position a word has in a corre-
lation. When two adjacent words have complementary functions
of the same Ic - for instance, word A has 5050 N$\underline{1}$ and word B
has 5050 N$\underline{2}$ - a 'product' is made and recorded in the form:

                 Word A   5050 N   Word B

This product is considered as one piece and can become
first or second correlatum in a wider correlation and is
therefore treated as though it were a single word, i.e., it
is assigned strings of Ic's which indicate its correlation-
al possibilities both with adjacent words and with adjacent
products already made. Single words, however, being vocabu-
lary items, have their strings of Ic's assigned a priori;
products, since they arise during the procedure, have to
be assigned their Ic-strings dynamically. The assignation
of specific Ic's to a product depends on:

a) the correlator responsible for the particular correlation;

b) the characteristics (Ic's) of the word (or product) which

makes up the first or the second correlatum.

The operational cycle that assigns Ic's to a product we call 'reclassification'.

The amount of data involved in an analysis of this kind is really enormous. Let us consider a sentence consisting of ten words, each of which has two different senses (S's). On an average 50 correlational indices are assigned to each sense of a word. Now, just to check the correlational compatibility of two adjacent words about 10,000 matching operations would be necessary; the matching procedure for all the words of the sentence would involve about 90,000 operations. On an average five products would result from the first 10,000 matching operations; each of them would be assigned about 50 correlational indices that represent the product's correlational possibilities to correlate with a another adjacent piece - either a word or a product. The procedure to match these five products with another piece would involve about 637,000 operations. If to this figures we add the number of operations necessary starting from level 3 (see p. 7) with all the products made in the immediately preceding levels (200,000), the total number of operations involved would come to 927,000.

The reclassification routine also involves a great number of operations of this kind: about half of the correlational indices a product is assigned depends on the corre-

lator responsible for that correlation; the other half de-
pends on the strings of indices which the two correlata of
the product have. According to the presence or absence of
specific indices in the strings of the first or second cor-
relatum, pre-established sets of indices are assigned to
the product; or sets of indices are assigned to the product
only if they are present in the strings of its two corre-
lata. The reclassification of each product would require
about 2,000 operations, which means 100,000 for the average
of 50 products in a sentence of 10 words, bringing the to-
tal of operations to over a million only for the matching
procedure. This would imply - for this part of the program
alone - processing times of the order of some seconds of
machine time if the most modern computer is available, or
of about an hour - at best - if the work is done with an
older model.

The amount of work and money involved in a procedure of
this kind made us try to find a quicker and more economical
way of handling correlational indices: as a result of our
efforts the Multistore system was developed.(Bibl. 1)

The basic idea of the Multistore consists in pre-estab-
lishing in a given area of the machine's central core as
many separate positions as there are correlators in the
system. The arrangement of these positions representing the
correlators orders them according to type. This assures
that at any point of the procedure each Ic is not used sev-

eral times and in different ways according to the diverse

data it contains, but only as one single item which by its

positional coordinates implies its various significations.

Moreover, the Ic's do not have to be compared one by one

with the Ic's of other adjacent words or products, but are

simply addressed to one and only one pre-established posi-

tion. Thus the mass of operations of comparison is avoided

and also the necessity to ascertain, after every successful

match, which items the matched Ic's represent is eliminat-

ed, because the very position of the matched Ic's immediate-

ly implies what they stand for. To establish whether two

Ic's are complementary and represent a correlation thus be-

comes the simple task of checking already present informa-

tion according to the rules of sequence, of correlational

function, and of correlator type, all of which are implicit

in the location of the markers which are being handled.

The Multistore can be represented as a rectangular area

divided into lines and columns. (see Fig. 1 below)
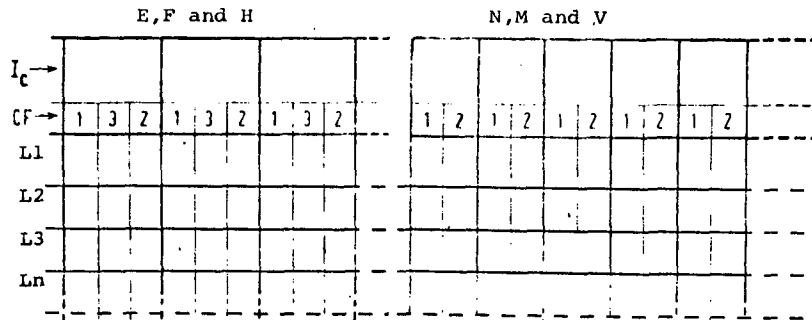
Every column is dedicated to one Ic and subdivided into two
subcolumns, if the Ic is of type N, M, or V (implicit); if
the Ic is of type E, F, or H (explicit), the column is di-
vided into three subcolumns.

The lines L1, L2, L3 etc. divide the area into levels. The
levels are determined by the succession of words in input.
Thus each level bears the number of the word it represents.
Every input word causes for each Ic in its Ic string the
insertion of a marker into the Multistore column correspond-
ing to that Ic; and the level of that marker in the Ic col-
umn corresponds to the input number and the position of
that word in the sentence. Thus all the markers inserted
for one word represent the correlational possibilities of
that word.

If, on the line of level 1, an Ic of the string represent-
ing correlator type N of the first word has caused the in-
sertion of a marker into the column corresponding to that
Ic and if an Ic of string N of the second word has caused
the insertion,into the line of level two, of a marker re-
presenting CF2 of the same Ic, this implicitly means that
with the same correlator a correlation is made containing
the first and the second word; this product No x, consist-
ing of word 1 (S 'a') and word 2 (S ,'b') is the product of
correlator No y and type N. This product belongs to level
2 and when it has been assigned a string of Ic's by the
appropriate rules of reclassification, it will be inserted

into the Multistore on the second level; this means that
it can enter into combinations only with those words that
belong to the immediately preceding level, or with products
which contain the words of the immediately preceding level.
Such a correlation, whenever it is made, would still belong
to the level of product x. In our specific case product x
could correlate only with an item of level zero, which does
not exist, because product x is on level two and already
contains word No. 1. Hence we can formulate a restrictive
rule to the effect that a product can be a potential second
correlatum in an N correlation only if its lower level is
larger than 1. The Multistore system lends itself to the
introduction of many such restriction rules.

When on a given level all products that have sprung from
the insertion of markers corresponding to the word of that
level have been reclassified,and the products originating
from that reclassification have, in turn, been reclassified
and have inserted their markers, and there are no more prod-
ucts to be reclassified, then the procedure inserts the next
word and thus begins the next level. This means that once
a subsequent word of the sentence has been inserted, all
preceding words and products become 'inactive' pieces, hav-
ing exhausted every possible attempt of correlation with
'active' pieces; from then on they represent merely latent
correlational possibilities with subsequent items. This
state of inactivity in the case of the last word of the sen-

tence determines the end of the analysis. At this point
the product (or products) that contains all words of the
sentence is called 'complete' and represents the hierarch-
ical structure of the sentence.

The first tentative program MP1 (Bibl. 1) was written
for use on a GE 425 computer and its main purpose was to
show the applicability of the Multistore system to corre-
lational grammar and to check the method of programming
based on 'significant addresses'.

The present program, MP2 (Bibl. 3) , is a revised and
enlarged version of MP1 written for use on an IBM 360/67
computer. On the basis of our previous experience it can
be considered an actual working tool.

Many solutions, as well as many restraints, depend on
the fact that under many respects it is a machine-orient-
ed program. The program is structured on a large area of
the central core, divided into lines and columns, whose
size is 528 x 330 bytes. Each line (330 all together)
consists of 528 bytes and is divided into two sections:
A and B. Section A contains all the data necessary to
define a line; section B consists of 496 bytes, that is,
of as many bytes as there are correlators operative in the
system. Each line specifies as permanent data a reclassi-
fication rule - whose definition is given in section A of
the same line - and the set of indices assigned by that
rule (bit 6). The relevance of the rule to a given product

is specified in the columns of section B.

A (32 bytes)   B (496 bytes)

1	
2	1st Cor. 2nd ****
3	
330	
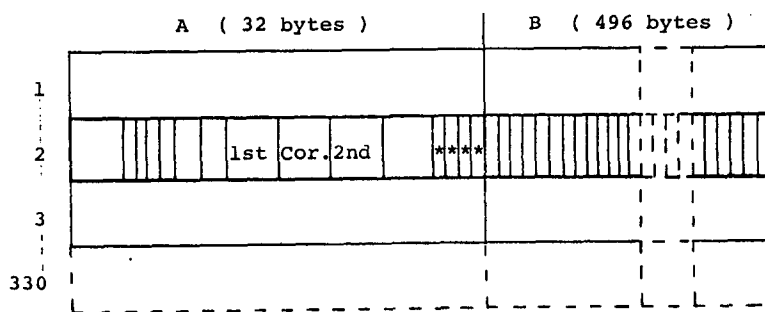
* reclassification rule

Fig. 2

Each byte of section B is divided into 8 bits as illustrated below.

0	
1	marker of CF3 (explicit correlator)
2	marker of CF2 (right-hand piece)
3	marker of CF1 (left-hand piece)
4	marker for special linguistic rules
5	reclassification rule marker
6	Ic assignation marker
7	.
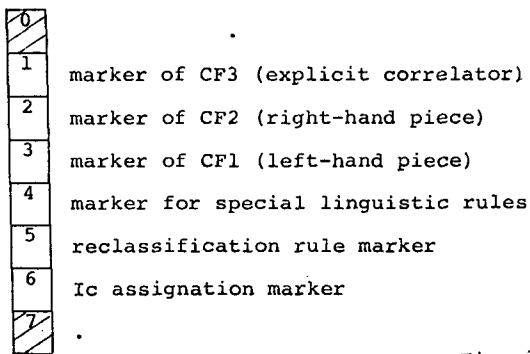
Fig. 3

Bits 1 to 4 are therefore used in the matching procedure, whereas bit .5 and 6 are pre-established data to be used in the reclassification routine.

Procedure

Each S-value of a word occupies one line of the Multi-
store area and its specifications are recorded in section
A of the same line. For each Ic contained in the string of
that S-value of the word, a marker is inserted , according
to the correlational function, in bit 1,2 or 3 of the cor-
responding byte of the line, that is, in the byte which
bears that Ic as label.

According to its function, a marker can be a left-hand
piece 'LH', and as such it is simply recorded, or a right-
hand piece 'RH', in which case, immediately after it has
been recorded, the column is searched for a complementary
and contiguous LH piece. If this is found, an indication
of product is recorded in the first free line of the Multi-
store; this address consists of three data: a) the address
of the line where the LH piece was found, which is recorded
in the area 'first correlatum';of the line of the product;
b) the address of the line where the RH piece was recorded,
which is recorded in the area 'second correlatum' and c)
the relative address of the column which characterizes both
both LH and RH pieces, which is recorded in the area 'cor-
relator'.

After the product's specifications are recorded, and
if there are no other LH pieces with which the RH piece in
hand can combine, the routine for the insertion of Ic's is
resumed. If the insertion of the next Ic of that S-value

— N sector —

| | W/P | S | 1st. | Corr. | 2nd. | | Rules | | | | | | | | | | | | | | | | |

527

* The position 6 in the Multistore area corresponds to correlator No y
in the same way as position 7 corresponds to correlator No z, and so
on.

INSERTION DIAGRAM

of the word causes a new product to be made, the procedure
is repeated and the product is recorded on the next free
line of the Multistore area. Only when all the Ic's of the
piece which has caused the production have been inserted,
the reclassification routine takes place, starting from the
first product newly recorded.

The information contained in the area 'correlator' of
the line containing the product's record gives the address
of the Multistore column dedicated to the correlator re-
sponsible for that product. The column is then searched ,
from the top down, for a bit 5 set ON (see Fig. 3 on p.10).
If it is found, this implicitly means that on the line to
which the bit belongs, there will be found the record of a
reclassification rule relevant to the product to be reclas-
sified. Section A of the same line contains the instruct-
ions concerning the assignation of the Ic's whose markers
are contained in bit 6 (see Fig.3). A bit 6 set ON implicit-
ly indicates either the column in which to check (if the
rule requires it) the record of the first or second corre-
latum(the addresses of which are recorded in section A
of the line in which the rule is recorded) for the presence
or the absence (as specified by the rule) of bits 1 to 3
set ON (which represent Ic's); or it may simply indicate
the place in which to insert a marker, i.e. a reclassi-
fication Ic, The routine for the insertion of reclassi-
fication markers is exactly the same as the routine for
the insertion of markers for words.

* Conditioned rule. $\underline{1}$ Unconditioned rule. # Check on lst correlatum.
$\overline{\underline{S}}$ Ic 4 CF1. @ Assign the string CF1 contained in the rule to the product.
$\overline{\underline{\%}}$ Assign the string CF2 contained in the rule to the product.

RECLASSIFICATION DIAGRAM

The analysis of the sentence is complete when the last
marker of the last word-sense has been inserted and there
are no further products to be reclassified or re-cycled.
At this point the output routine starts. Three different
kinds of output are produced:
a) a list of all the products made in the course of the
   analysis of the sentence;
b) a list of all Ic's assigned to each product during the
   reclassification routine ;
c) a graphic representation of the hierarchical structure
   of all 'complete' products (that is, containing all
   words of the sentence). This structure is equivalent to
   a tree structure with words at the terminals and corre-
   lators at the nodes. (see Appendix)

This is a general outline of the procedure of combina-
tion, production, reclassification and output. In addition
to that there are several routines which meet special re-
quirements. A special rule, for instance prevents specific
RH pieces from becoming eligible LH pieces once a certain
correlation - which contains them as RH pieces - has been
made. A word like "LITTLE", for instance, in its function
as a quantifier, once it has been correlated with the de-
finite article and made the product "THE//LITTLE" cannot
become LH piece in the correlation:

LITTLE // HE KNOWS

The indication 'discard' on print-out type 'a' - i.e. on
the list of all the products made during the analysis -
will show that "LITTLE" is no more available as LH piece
for any other correlation.

   Another restraint concerns some 'complete' products
which, though grammatically correct, cannot be accepted
as interpretation of the sentence. For instance, in a
sentence like:

                    THEY // WERE READY

the structure which takes "WERE" as subjunctive is not
acceptable, since it would require something else - an
"IF" or "I WISH" etc. - to precede. In cases like this
the indication 'non-sentence' appears in print-out type
'a'.

   A set of special routines serves the purpose of rec-
ognizing idiomatic expressions. When one of them is rec-
ognized, inserted in the Multistore and reclassified -like
any other product - the indication 'idiom' is printed on
print-out type 'a'.

   The whole program, including the Multistore area and
buffers, is contained in the central core of the machine
and occupies about 200 K. The system accepts as input,sen-

tences of up to 16 words - a limit fixed in accordance with
the average length of sentences in scientific texts (Bibl.5)
and ample enough to allow any type of syntactic structure.
Processing-times for 10-word sentences are about 1-1.5 sec-
onds. Our present vocabulary is limited to 150 words for
reasons of punched card maintenance. However, it could be
enlarged without affecting the program.

The Multistore parser was developed for the automatic
analysis of English sentences on the basis of correlation-
al grammar, but it is in no way limited to this kind of
grammar. Actually, by changing the input parameters, the
symbols of the rules and the matching operations it could
be used to handle the data of any kind of predictive gram-
mar; neither are its dimensions critical; the Multistore
area could be reduced or enlarged simply by altering the
parameters in accordance with the storage capacity avail-
able in the machine.

Bibliography

1   'Multistore': A Procedure for Correlational Analysis
     (E.v.Glasersfeld, P.P.Pisani, J.B.Burns), Informal
     Report T-10, Automazione e Automatismi, vol. IX, No.2
     Milan, Italy, 1965

2   Automatic English Sentence Analysis (Glasersfeld,
     Pisani, Burns, Notarmarco, Dutton), Final Report T-14
     Grant AF EOAR 65-76, IDAMI Language Research Section,
     Milan, Italy, 1966.

3   The Multistore System MP-2 (E.v.Glasersfeld and P.P.
     Pisani), Scientific Progress Report, Grant AFOSR 1319-67
     Georgia Institute for Research, Athens, Georgia, 1968.

4   The Multistore Parser for Hierarchical Syntactic
     Structures (E.v.Glasersfeld and P.P.Pisani) Grant
     AFOSR 1319-67, Georgia Institute for Research, Athens,
     Georgia 1969 (paper submitted to Communications of ACM)

5   Computational Analysis of Present-Day American English
     (Henry Kucera and W.Nelson Francis), Brown University
     Press, Providence, Rhode Island, 1967

Input Sentence: ...

(Print-out type a)

A P P E N D I X , I

Complete Parsing

II

S CC15	W CCC5 01 VS	0550 E	W 0007 0101 /	C4 09		
P CC16	W CCC6 01 VS	4420 A	W CCC7 02 CN	C6 09		
P CC17	F 0009	2220 A	P 0019	C4 09		
P CC18	P CC17	4010 W	W 0003 01 42	C3 09		
P CC19	W CCC6 01 VS	0550 E	W 0009 02 42	C6 09		
P CC20	F CC19	0550 E	W 0008 02 46	C3 09		
P CC21	F 0012	C550 E	W CCC5 02 42	C4 09		
P CC22	F 0009	2220 N	F 0019	C4 09		
P CC23	W CCC3 01 FS	8571 V	P 0021	C3 09		
P CC24	F 0022	4010 N	W CCC3 01 43	C3 09		
P CC25	W CCC2 C4 FS	4010 A	F 0023	C2 09		
P 0026	W CCC2 03 FS	4010 N	P 0022	C2 09		
P CC27	W CCC2 01 FS	4010 A	F 0023	C2 09		
P CC28	W 0001 01 41	2250 N	P 0025	C1 09	COMPLETE	

(Print-out type c)

```
178540
```

```
P C28   +-------------------------------------------------------2250N-+
          .                                                         .
P C26     .        +------------------------------------------4010N-+
          .        .                                                .
P C23     .        .        +----------------------------------8571V-+
          .        .        .                                        .
P C21     .        .        .        +------------------------0550F-+
          .        .        .        .                             .
P C12     .        .        .        +-------7016A-+
          .        .        .        .             .
P C11     .        .        .        .      +-5210A-+
          .        .        .        .      .      .
          -----+------REA.FR  11  53, HIS CP  PRI 1A  HCC 53  CN      TCA 7
```

IV

P C25 P C27 P C23 P C21 P C12 P C11

178542