DOCUMENT RESUME

ED 024 930 AL 001 582

By- von Glasersfeld, Ernst; Pisani, Pier Paolo
The Multistore System: MP-2
Georgia Inst. for Research, Athens.
Pub Date Nov 68
Note- 72p.
EDRS Price MF-$0.50 HC-$3.70
Descriptors- *Computational Linguistics, Computer Programs, English, Form Classes (Languages), Kernel Sentences, Linguistic Patterns, Machine Translation, Phrase Structure, *Programing, Semantics, *Sentence Structure, *Structural Analysis, Structural Grammar, *Syntax
Identifiers- *Correlational Grammar, Parsing

The second version of the Multistore Sentence Analysis System, implemented on an IBM 360/65, uses a correlational grammar to parse English sentences and displays the parsings as hierarchical syntactic structures comparable to tree diagrams. Since correlational syntax comprises much that is usually considered semantic information, the system demonstrates ways and means of resolving certain types of ambiguity that are frequent obstacles to univocal sentence analysis. Particular emphasis is given to the "significant address" method of programming which was developed to speed up the procedure (processing times, at present, are 0.5-1.5 seconds for sentences up to 16 words). By structuring an area of the central core in such a way that the individual location of bytes becomes significant, the shifting of information is avoided; the use of binary masks further simplifies the many operations of comparison required by the procedure. Samples of print-out illustrate some salient features of the system. (Author/MK)

Natural Language Research

THE MULTISTORE SYSTEM

MP-2

Ernst von Glasersfeld
Pier Paolo Pisani

GEORGIA INSTITUTE FOR RESEARCH
Athens, Georgia

GRANT
AFOSR 1319-67

MP-2
November, 1968

THE MULTISTORE SYSTEM

MP - 2

Ernst von Glasersfeld
Pier Paolo Pisani

Scientific Progress Report

GEORGIA INSTITUTE FOR RESEARCH
711, C & S Bank Bldg.
Athens, Ga. 30601

ABSTRACT

The report describes procedure and machine program of the second version of the Multistore Sentence Analysis System implemented on an IBM 360/65. Using a correlational grammar (described in previous reports) the system parses English sentences and displays the parsings as hierarchical syntactic structures comparable to tree-diagrams. Since correlational syntax comprises much that is usually considered semantic information, the system demonstrates ways and means of resolving certain types of ambiguity that are frequent obstacles to univocal sentence analysis.

Particular emphasis is given to the 'significant address' method of programming which was developed to speed up the procedure (processing times, at present, are 0.5-1.5 sec. for sentences up to 16 words). By structuring an area of the central core in such a way that the individual location of bytes becomes significant, the shifting of information is avoided; the use of binary masks further simplifies the many operations of comparison required by the procedure.

Samples of print-out illustrate some salient features of the system.

ACKNOWLEDGMENTS

We should like to express our thanks to the University of Georgia, and in particular to the heads of the Office of the Vicepresident for Research, for the interest and the understanding they have shown for our effort and for having made it possible for us to have continuous access to the University's Computer Center.

We also wish to thank the many research groups in the United States and elsewhere who have sent us reports of their work; a selected bibliography of papers dealing with problems relevant to natural-language analysis will be published with our Final Report.

CONTENTS

	page
Abstract	
Acknowledgments	
Table of Contents	
The Multistore Parser	
Introduction	1
General Description	6
General Data Flow	12
Permanent Data	14
Input Data	20
Multistore Program	22
Output	43
Bibliography of Relevant Reports	46
Appendices	
Explanation of Print-out	I-a
Sample of Structure Display	I-b
Sample of Prepositional Analysis	I-c
Sample of Operative Restraint	I-d
Operative Vocabulary	II

The research reported in this paper has been sponsored by THE AIR FORCE OFFICE OF SCIENTIFIC RESEARCH under Grant AFOSR 1319-67, Information Sciences Directorate.

Introduction

In its general conception the new version of the Multistore program is based on MP-I, which was described in a previous report of the group (ILRS T-10, January 1965). The details of the procedure, however, have undergone considerable modification. On the one hand, this was necessary because MP-I had been written for use on a GE 425 computer, while the machine we have been using since our transfer to the United States is an IBM 360/65 (University of Georgia Computer Center) whose technical characteristics made necessary a rather far-reaching reorganisation of the procedure; on the other hand, as the program had to be rewritten in any case, we took this opportunity to incorporate in it some of the improvements and new ideas that had been developed during the period of experimentation with MP-I.

Description and explanations - from various angles - of the Correlational Grammar underlying this application of the Multistore system can be found in previous reports (cf. Bibliography) and we, here, limit our exposition to a very brief outline of those dispositional aspects of the grammar that are indispensable for an adequate understanding of MP-II.

In principle, the Multistore system can be used as a parser with any kind of predictive grammar that supplies the items of a given vocabulary with exhaustive and univocal indications as to their syntactic combinability in sentences of a natural language. Most grammars classify vocabulary items according to their general syntactic behaviour (which leads to relatively few but crowded classes, e.g. nouns, verbs, adjectives, etc.) and then proceed to subdivide according to the specific or "exceptional" behavior of certain items or groups of items. One might call this the botanist's approach; as with trees or flowers, it is eminently useful with the word items of a natural language - provided that the principal purpose of the effort is the description of these items. But if the purpose is the interpretation of sentences, i.e. of combinations of items, then a classification's usefulness and efficiency depends on how accurately it explicates and displays the individual combinatorial behaviour of the items involved.

Correlational grammar was designed specifically for this second purpose. It deviates from traditional grammar in that it characterises the word-items (i.e. single words and phrases) exclusively in functional terms, and not according to their phonological or morphological aspects.

This characterisation in functional terms is achieved, on the one hand, by a minute and rigorous discrimination of syntactic functions (called 'Correlators') and, on the other, by assembling each individual word-item's characterisation in the form of a string of indices (Ic's), each of which indicates the item's specific possibility of functioning as one term (either first or second 'Correlatum') of one particular syntactic combination.

In fact the individual combinatorial behaviour of a word-item is, in many instances, determined by characteristics which would not be regarded as 'syntactic' in the traditional acceptation of that word. This, of course, depends on how one defines the term 'syntactic'. In the context of correlational grammar, a characteristic is called 'syntactic' when it determines a word-item's eligibility as correlatum of a specific correlator; and it is called 'semantic' when it determines the compatibility of that word-item with another word-item within a given correlation (*).

The Ic-string of a word can be considered its grammatical classification; but whereas in traditional grammar, when a given word is classified as 'verb', this implies that the word can be used as second term in subject-verb constructions, in correlational grammar there is no such generic term for the

---

* cf. Jehane Burns, Bibl. N° 9.

word, but instead a more or less numerous group of Ic's in the word's Ic-string, which specifies exactly the types of subject-verb correlations the word can enter into.

Once a large vocabulary is thoroughly classified by Ic-assignation, it will obviously be possible to group the vocabulary's items according to similarities in their Ic-strings, and thus to arrive once again at syntactic word classes. But since our project was undertaken as a feasibility study (to explore the applicability of the correlational method to the analysis of English sentences), we have, so far, had neither the time nor the need to prepare a vocabulary large enough to serve as basis for a distributional examination of syntactic characteristics.

The suitability of the correlational grammar for automatic sentence analysis depended - in our view - on the answers to three questions:

1) Can a correlational grammar account for all the syntactic structures found in ordinary English sentences and can it satisfactorily recognise and interpret the structure of sentences when it is used in a recognition procedure or parser?

2) Can such a parser be improved by the incorporation of semantic data?

3) Is it possible to program an automatic correlational parser so that it will yield the analysis of an average English sentence in a reasonably short processing time - i.e. within a few seconds rather than minutes ?

Question (1) was, in principle, answered by MP-I. Although output from that program showed that the grammar with which it was working was not sufficiently differentiated in certain areas, it clearly demonstrated that the correlational system could yield an efficient recognition procedure and, owing to its essential open-endedness, was capable of any desired degree of refinement (without basic changes in its structure).

Refinement of the grammar has been going on continuously throughout the effort, and the operational grammar of today, consequently, is much more efficient and reliable than the one implemented in MP-I. This process of refinement - which often requires painstaking studies of specific types of syntactic structure - has not yet reached an even level of sophistication in the entire gamut of structures possible in the English language. This is due partly to the inevitable delay in the machine implementation of the linguist's latest advances (re-writing of rules, re-punching of cards, etc.); and partly it is due to the fact that much of what, originally, had been expected to be resolvable only by the introduction of semantic data, has turned out to be within the reach of correlational syntax. This is so in the case of certain prepositional relations (cf. the treatment of 'pseudo-ambiguities' outlined in "An Approach to the Semantics of Prepositions", Bibl. N°5) and instances of the elimination of pseudo-ambiguities in the use of the preposition "by" can be seen in the examples of output (Appendix I-c) other instances of correlational syntax handling problems which previously had been considered 'semantic' are the resolution of pseudo-ambiguities in the area of predicative adjectives(*)(e.g. "John is easy to please","John is eager to please","John is likely to go", and "John is kind to go") and in the area of phrase governing verbs (**)(e.g. "he works the land to live","he promised us to come","he forced us to come", etc.).

Given this extension of syntax at the expense of semantics, our approach to the problems of semantics (question 2, above) has been somewhat modified. We have become convinced that, at the present stage in the development of our parser, it is important to exhaust the possibilities of syntactic ambiguity

---

* cf. Some Adjective Classes Derived from Correlational Grammar, Bibl. N° 7.

** A paper on this specific subject is in preparation.

resolution before we introduce any system of disambiguation based on semantic factors. This now seems advisable, not because we posit an operational precedence of syntax over semantics (such a precedence is certainly not observable in the analysis procedure employed by the human sentence interpreter), but because a small research group cannot possibly deal with both areas simultaneously; moreover, a special study investigating the addition of semantic control mechanisms to the Multistore system is being carried out under a collateral research project (cf. Bibl. N^{os} 9, 10).

On the basis of MP-I, the third question (see above) could be answered affirmatively, but with a reservation. The processing times for short sentences were, indeed, in the range of a few seconds; but the system was so drastically limited as to vocabulary, sentence length, and space for grammar improvement, that it was legitimate to doubt whether the processing speed could be kept at a reasonably high level once the system's scope was increased to realistic proportions.

With the reorganisation of the reclassification routines (the most complex and slowest part of MP-I) this doubt has been obviated: Reclassification in MP-II, thanks to the superposition on the Multistore area (cf. 6.00-6.52, below), has become the fastest part of the program, and the effect of sentence length on the processing time is therefore approaching its minimum.

General Description and Outline of the Procedure

The Multistore Parser accepts English sentences consisting of words that are contained in the system's vocabulary. In the vocabulary each word is characterised by strings of indices (Ic's) which indicate its possibilities of combining with other words or phrases to form syntactic structures (Correlations). Each Ic specifies one syntactic function by means of which the item bearing that Ic can be correlated to another item, thus forming a correlation; the Ic also specifies the item's place (Correlational Function) in the indicated correlation.

The parsing of an input sentence is effected by matching the complementary Ic's of its words and - once words have been combined - correlations. Two Ic's are considered matching when they indicate the same correlator but different and complementary correlational functions.

The present version of the system works with approximately 300 syntactic functions, and the individual word items, therefore, have long strings of Ic's - averaging about 30-40. The number of matching operations (considering not only the Ic's of the single words but also those of the word combinations possible within a sentence) is, consequently, very high.

The Multistore procedure was devised to reduce the number of operations and to make them as fast as possible. It does this, on the one hand, by spreading a temporal sequence over a spatial area, thus making steps contemporaneous; and, on the other, by giving each Ic a fixed position in that area (significant address), so that the matching operations can be carried out without the shifting of information. Although a computer's memory is usually considered to be linear, addresses can be arranged so as to represent any kind of area. The Multistore can best be visualised as a rectangular area with horizontal lines and vertical columns. Each column represents one specific Ic; the columns are grouped in eight sections, six for the different correlator types (cf. 1.23), one for recognition indices (cf. 1.25) and one, the first on the left, for the specification of the item that occupies a given line (cf. 3.17 and fig.3). Each line represents an element of the sentence, either word or word combination.

As the sentence to be analysed is input, the first word occupies the first line of the Multistore and its Ic's are recorded (by setting ON certain bits) in the bytes that constitute the intersections of this first line with the columns representing the relevant Ic's. The correlational functions of the Ic's are represented by the configuration of the bits that are set ON within the Ic's byte; this portion of the byte then constitutes a 'marker' for the particular correlation index read in the word's Ic-string.

When the second word is input, it occupies the second line of the Multistore. While its Ic's are being recorded, those whose function indicates the word as a possible right-hand item in a correlation made with the preceding item, triggers a search of the column (into which its marker is being inserted) to see whether the complementary function was marked for a preceding item. If such a marker is found in a contiguous position (cf. 4.03), it means that a correlation is possible (subject to the checks explained in 4.00), and it is immediately recorded as a 'Product' on the next free Multistore line, that is to say, in the part of the line reserved for the specification of items; it should be noted that the product is "automatically" specified by the place of its constituents: the right-hand piece is the item whose Ic-marker was being inserted; the left-hand piece is the item represented by the line on which the complementary function was spotted; the correlator is indicated by the Ic-number of the column in which the combination was made. (Note that in correlations with normal word order the left-hand item is the first correlatum, the right-hand item the second correlatum; in correlations reflecting an inversion of the word order in the sentence, the roles of 1st and 2nd correlatum are inverted, but the procedure of correlating them remains the same; cf. 1.23 and 4.02).

In this way the matching operation, instead of involving the scanning of long strings of indices, is reduced to a simple binary check; and the same binary check yields, whenever a correlation is actually found to be possible, the data required to characterise the product.

Regardless of whether or not a product results from the scanning of the column, the procedure goes on to the next Ic until it reaches the end of the string of the word-item in hand.

When the last Ic of the item has been dealt with, the Reclassification procedure sets in. Reclassification is the process of assigning correlation indices to products, i.e. the process of supplying a word combination with those and only those Ic's that reflect its correlational possibilities with regard to other words or word combinations. It is the most complex part of the system and it is here that MP-2 shows the most important conceptual advance in comparison to MP-1.

The linguistic aspects of reclassification and the general principles underlying the formulation of the rules which govern the reclassification of products correlated by the individual correlators have been discussed in a previous report (ILRS T-14, Section VI); a list of the principal types of rule used in the system will be found under 6.20 ff. below).

From the point of view of the procedure each correlator has an individual set of Reclassification Rules - and this set we call Reclassification List. Some of these

Rules are unconditional, in the sense that every product made by the specific correlator will receive the string of Ic's they assign; others are conditioned, i.e. they assign a string of Ic's only if, say, the first correlatum of the product has a certain characteristic.

Many of these Rules figure in more than one List. Since it would be wasteful of both storage space and operational time to keep voluminous tables of Lists and Rules outside the computer's work area, a method was devised to incorporate the relevant data in the Multistore area.

As the Multistore area is structured in columns, every one of which is dedicated to one correlator, each individual correlator's column is now used to record that correlator's reclassification List; and the lines of the Multistore area are used to record the Rules in such a way that each Ic they assign is indicated by another kind of marker at the intersection of that line with the column dedicated to that Ic (cf.3.17-3.19, and Fig.3,7).

The reclassification procedure begins with the first product recorded in the left section of the Multistore area during the insertion and combination cycle that has just ended. The correlator responsible for this product determines the column in which the relevant reclassification List will be found. This column is then scanned for Rule markers (i.e. a certain configuration of bits, cf. 6.03) which indicate that the line on which the marker is found contains the details of a Rule to be applied to the product in hand.

Whenever a Rule marker is found, the scanning of the column is interrupted and the line is examined from left to right. In the first section of the line there is the specification of the Rule, indicating the correlational function of the Ic's to be assigned by the Rule and the conditions of their assignation. If the product satisfies the conditions, the main part of the line is scanned, i.e. the part which crosses ~~the~~ the Ic-columns of the Multistore area, and a sign (second type of marker) is found in those columns that represent an Ic to be assigned by the Rule. In these columns, then, an Ic-marker is inserted at the intersection with the line of the product that is being reclassified, and this insertion has the same effect, and is followed by the same steps, as the insertion of an Ic-marker from the string of an input word (a reclassified product, thus, is treated in exactly the same way as an input word).

The operational path that starts with the identification of an Ic in the line of a reclassification Rule (which is being examined to determine the Ic's that are to be assigned to a newly made product) may thus lead to the creation of a new product; and this is why, in the program, combination routines and reclassification routines are closely interwoven.

At the end of every such 'detour', however, the scanning of the Rule's line continues. When the end of the line is reached, the procedure returns to the column of the List which governs the reclassification of the product in hand, identifies the next Rule marker, and follows the path entailed by the conditions and signs encountered in the line that contains the indicated Rule. And so it goes on, until there are no more new products recorded in the left-hand section of the Multistore area, i.e. no products that have not been reclassified.

Only at this point does the next item enter into the combination routine; this new item can be either a separate sense of the same word, or the next word of the input sentence.

When the last word has been dealt with, output begins and those products recorded that contain <u>all</u> words of the sentence are printed out and their correlational concatenation, i.e. their syntactic structure as expressed in terms of correlations, is graphically displayed. This display, which indicated the words, the correlators that connect them, and the hierarchy of correlations, is in fact a binary tree-structure and constitutes the parsing of the sentence.

General Data Flow

1.00 The permanent data base of the Multistore Parser consists of vocabulary and grammar. The words contained in the vocabulary are represented by punched cards and in a disc file. The grammar, on the other hand, is not pfysically represented in any one place; parts of it are represented by the strings of correlation indices of the vocabulary items, parts by the Reclassification Lists and Rules, and parts are built into the various combinatorial modes of the correlation procedure.

1.01 The general structure of the system can be set out as follows:

A - Permanent Data		
	Vocabulary	(1.10-1.26)
	Idiomatic Phrases	(1.30-1.31)
	Reclassification Lists	(1.40-1.44)
	Reclassification Rules	(1.50-1.54)
B - Input Data		(2.00-2.22)
C - Multistore Program		(3.00-7.50)
D - Output		(8.00-8.31)

(Note: The decimal numbers refer to the relevant paragraphs of the text.)

For an illustration of the general data flow, see Fig 1, next page.

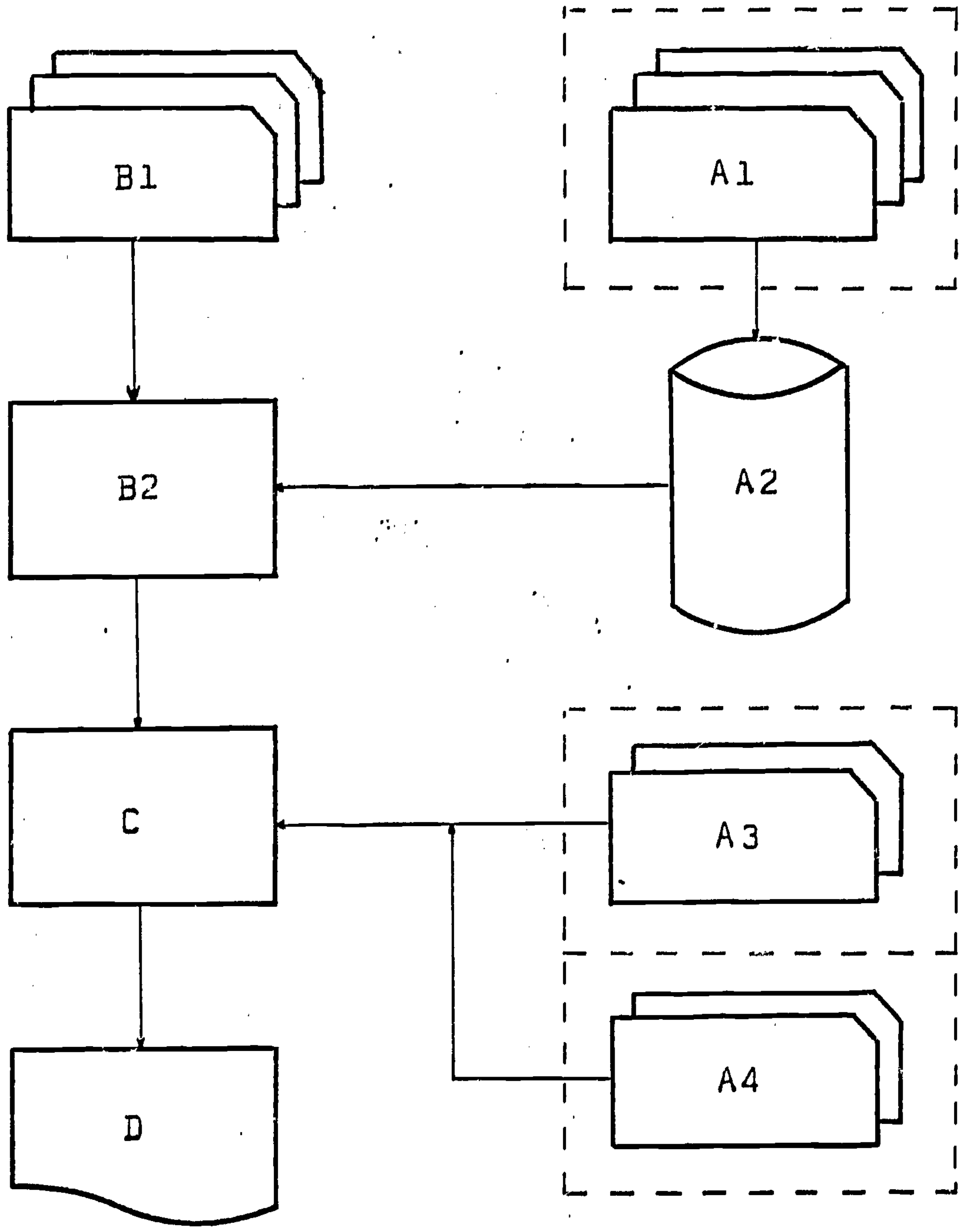


Fig. 1

A1 - cards type 30 and 31, input to establish vocabulary;

A2 - disc file containing the vocabulary;

A3 - cards type 10, input of reclassification lists;

A4 - cards type 20, input of reclassification rules;

B1 - cards type 40, input sentence (words);
cards types 50, 51, 60, punctuation marks;

B2 - dictionary look-up;

C - Multistore program;

D - output, print-outs types a, b, c.

A - P e r m a n e n t   D a t a

1.10 The vocabulary contains all word-items the system can handle; consequently the system can analyse sentences that are composed exclusively of words contained in this vocabulary. (Since the system was designed for experimentation and research, no provision was made to enable the procedure to signal, by-pass, or otherwise deal with words that are not found in its vocabulary.)

1.11 The vocabulary is stored on a disc file (Data Cell). The data it contains are recorded on the disc file by means of punched cards that are input previous to the start of an analysis operation.

1.12 The vocabulary entry for one word-item consists of data which are written into the disc file by means of two different types of punched card: S-cards (formerly head-cards) and Ic-cards.

1.13 An S-card (punched card type 30) contains the following data:

- Vocabulary Number (cf. 1.14)
- S-Number (cf.1.15)
- G and T code (cf. 1.16)
- the word in letters (cf. 1.17)

1.14 The Vocabulary Numbers reflect the alphabetic order of words in the vocabulary, but they are not a continuous sequence since, in order to allow for the insertion of new words (expansion of the vocabulary), a regular interval of 20 numbers was left between the words chosen for the initial vocabulary; some of these numbers have since been occupied by subsequently added words.

1.15 The individual S-Number of an item is a two-digit code number which distinguishes the different senses (meanings) of the word, regardless whether the difference is semantic or syntactic. Thus there will be, for instance, three vocabulary entries (with different S-numbers) for "can":

can (S= 01) - modal auxiliary
can (S= 02) - verb (= to pack in cans)
can (S= 03) - noun (=container)

(Note: a semantic split of the modal auxiliary "can" - a ability and b possibility - would be necessary for translation into certain languages but has not yet been implemented in the vocabulary.)

The code numbers (01, 02, etc.) do as yet not indicate specific types of sense or function; they merely distinguish homographs from one another. In a future version of the program we plan to integrate this S-code with a comprehensive grammatical and semantic code of 8 digits.

1.16 The G- and T-code of the item corresponding to a given S-number is a 6-digit code number divided into two areas: a 2-digit area G, and a 4-digit area T.

At present only the area G is being used. It contains a rough, temporary grammatical classification, introduced provisionally for the purpose of testing reclassification routines conditioned by data in that specific coding area.

The definitive classifications to be coded in the six digits of the G&T area will be developed on the basis of research that is still under way.

1.17 Each S-card bears the alphanumeric representation of the word it refers to. The maximal number of letters for one word is 12 (longer words have to be truncated or abbreviated).

1.20 For each S-card there is a set of Ic-cards (punched cards type 31) which bears the Ic-strings of the item corresponding to the S-card. The Ic's represent the item's correlational possibilities and are divided into separate strings according to the correlator type and the correlator function they indicate.

1.21 A Correlation Index, or Ic, consists of 6 digits and contains three different data:

1) the correlator number (4 digits), i.e. the number of one individual correlator (syntactic function); e.g. 4010, which is a verb//object relation; or 0245, which is the relation of 'spatial proximity' expressed by the preposition "by".

2) the correlator type (1 digit); the correlator type indicates certain basic characteristics of the correlations for which the correlator is responsible (cf. 1.23).

3) the correlational function CF (1 digit), i.e. the indication of the items place in a correlation, which can be:

CF = 1	-	1st term, or correlatum, of the correlation;
CF = 2	-	2nd term, or correlatum, of the correlation;
CF = 3	-	the correlator.

1.22 In the sentence "see London by night", for instance, the operative correlational functions of the items are:

"see": CF1 of correlator 4010 (verb//object);
"London": CF2 " " " " " ;
the correlator (4010 CF3) responsible for the correlation "see London" is implicit;
"see London": CF1 of correlator 0241 (activity// specification of lighting, cf. Appendix I-c);
"by": CF3 of correlator 0241;
"night": CF2 " " " ;
the correlator (0241 CF3) responsible for the correlation "(see London) by (night)" is explicit, that is to say, it is expressed by a specific word, namely "by".

1.23 The coding of correlator types is as follows:

Type N: implicit correlator; the correlated items (correlata), in the word order of the sentence, have functions:

CF1 + CF2 respectively;

Type M: implicit correlator; the correlata's functions are inverted in the word order of the sentence:

CF2 + CF1

Type V: implicit correlator with obligatory comma; the correlata's functions follow the word order and the correlata are separated by a comma:

CF1 + , + CF2

E and G: explicit correlators; the correlata's functions, in the word order of the sentence, have the sequence:

CF1 + CF3 + CF2

<u>Note</u>: Types E and G differ only in the partial combinations that lead to the correlation; in type E the item with CF1 is combined with the item bearing CF3 to form a 'Semiproduct' (cf. 4.11 ff) previous to combination with CF2; in type G the semiproduct is formed by CF3 and CF2 previous to combination with CF1; this second way of combining the three elements was introduced for experimental reasons, but is not operative with the present data base.

Type F: explicit correlator; the correlata's functions, in the word order of the sentence, have the sequence:

CF3 + CF2 + CF1;

<u>Note</u>: the semiproduct CF2 + CF1 is formed previous to combination with CF1.

1.24 The combinations of correlator type and correlational function that characterise different Ic-strings (cf. 1.20) are as follows:

N1, M1, V1, E1, G1, F1,

N2, M2, V2, E2, G2, F2,

E3, G3, F3.

1.25 Recognition Indices (cf. 3.14), although they are not Ic's, are included in the string N1. They are distinguishable from Ic's by the fact that their numbers are beyond the range of correlator numbers and therefore refer to a different section of the Multistore area (cf. 6.20 ff).

1.26 At present the average number of Ic's pertaining to one S-item is 28; the average number of Ic's on one Ic-card is 5.65. The word-item with the least Ic's has 3 Ic's and 3 Ic-cards; the item with the most has 136 Ic's and 22 Ic-cards.

1.30 Idiomatic Phrases, i.e. fixed word combinations whose meaning as a whole is not equivalent to the meaning of any syntactic structure composed of the same individual words, are treated like word-items in the vocabulary. Like single words, they are represented by an S-card and a set of Ic-cards.

1.31 A special routine, which runs parallel to the correlation procedure, spots them in the input sentence and brings them into play as vocabulary items the moment the last word of the idiomatic phrase has appeared in input.

1.40 Reclassification Lists.

For each correlator-number there is a set of one or more cards (depending on the amount of data concerning the particular correlation) which contain the code numbers of the Reclassification Rules relevant to products made by that correlator.

1.41 Since the reclassification data (Lists and Rules) represent the systems syntax and are, therefore, subject to frequent correction during the experimental stage of the work, they are not permanently stored on a disc or similar storage device, but are kept in the form of punched cards which are input together with the analysis program.

1.42 The List-cards (punched cards type 10) contain the following data:

- a label, which is the Ic whose products are reclassified by means of the reclassification Rules indicated in that List;
- a set of Rule-numbers which indicate the relevant Rules.

1.43 In the system, the data contained in the Lists are inserted into their specific positions in the Multistore area. The Ic that labels a given List also determines the Ic-column into which that List's data are inserted; the Rule-numbers determine the lines (of that same Ic-column) where the individual Rule-markers are inserted (cf. 6.02 ff).

1.44 There are as many Lists as there are correlators in the system. At present the system works with 266 correlators and the same number of operative Lists. The number of Rules indicated on one List varies between 1 and 50, with an average of approximately 20.

1.50 <u>Reclassification Rules</u>

The reclassification Rules that have to be applied to the products of one specific correlator are indicated by their code numbers on the reclassification List (cf.1.42) of that Ic; the specifications of the Rules, however, are fed into the system separately, on Rule-cards. This division of data was found to be economical because many Rules apply to the products of more than one Ic.

1.51 Reclassification Rules are kept in the form of punched cards which are input together with the analysis program (cf. 1.41).

1.52 For each Rule there is a set of one or more punched cards (depending on the amount of data involved in the particular Rule). These Rule-cards (punched cards type 20) contain:

- a label, which is the code number of the Rule;
- a string of one or more Ic's which specify the assignations to the product;
- indication of the correlator type and the correlational function to which the Ic's of the string refer;
- the conditions of assignation (cf. 6.20 ff).

1.53 In the system, the data contained in the Rule-cards are inserted into their specific positions in the lines of the Multistore area (cf. 6.02 ff).

1.54 At present the system works with 270 reclassification Rules, which assign from 1 to 26 Ic's (average approximately 6).

B - Input Data

2.00 The input to the system consists of English sentences. The system will process any sentence, provided that:

1) it consists of words that are contained in the operational vocabulary;
2) it does not exceed the length of 16 words;
3) it does not contain punctuation marks other than comma, full stop, and question mark.

2.01 The upper limit of sentence length was provisionally fixed at 16 words. The display of a correlational sentence structure with 16 word terminals can just be fitted on one page of print-out without cutting out any of the data required for immediate visual checking. Since the purpose of the program is above all experimental, and since 16 words are easily enough to exemplify all possible types of syntactic construction, this seemed a reasonable limitation - all the more as the sentence-length distribution in scientific writings has its absolute peak between the sentence-lengths of 16 and 19 words (*). The analysis program would require only slight changes if one wanted it to process longer sentences.

2.10 The sentences to be analysed (input sentences) are manually composed of Input-cards, one for each word of the sentence and one for each punctuation mark contained in it.

2.11 Input-cards representing words (punched cards type 40) contain the following data:

- vocabulary number of the word;
- the word in letters;
- the S-numbers of the different senses of the word;
- and their individual G-codes.

2.12 The input-cards, type 40, serve to call up from the vocabulary the data (cf. 1.12-1.22) concerning the words of the input sentence.

2.13 Owing to the fact that the Data Cell units of the computer we have been using were for a long time inaccessible, we have been running the Multistore program without them; that is to say, the vocabulary is kept in the form of punched cards, and the sentences that serve as input for the analysis procedure are composed manually.

2.14 The word-cards relevant to the input sentence are, therefore, input after the sentence and the input-cards call up the data from these. The switch-over to the disc file, however, is fully prepared.

2.20 Input-cards representing punctuation marks are independ-

(*) cf. Computational Analysis of Present-Day American English, by H. Kucera and W. Nelson Francis, Brown University Press, 1967.

ent cards (punched cards types 50, 51, 60) and must be inserted during the composition of the input sentence and into those places between the words where the punctuation marks actually occur.

2.21 The punctuation marks foreseen by the present program are:

- fullstop (punched cards, type 50);
- question mark (punched cards, type 51);
- comma (punched cards, type 60).

2.22 The punctuation cards do not refer to items in the vocabulary; they operate as switching signals and have a direct effect on the relevant subroutines of the correlation procedure.

C - Multistore Program

3.00 The Multistore program implements the correlational parser, that is to say, it analyses the input sentence as to the correlational structures contained and it displays in its output the 'complete' structures, i.e. those structures that embrace all the words of the sentence.

3.01 The analysis is based partly on the information concerning the correlational possibilities of the single word items of the sentence (Ic-strings, cf. 1.20 ff) and partly on the rules of word-order in English sentences (incorporated in the 'Modes' of combination, cf. 4.21-4.23).

3.02 The procedure can be divided into a number of different sub-procedures:

- input of the sentence;
- combination procedure (producing correlations of words and word-combinations);
- reclassification procedure (assigning correlation indices to word-combinations);
- restraints (preventing or eliminating certain combina-

tions which are generally possible, but, in the particular case, impossible owing to some feature of the given sentence);

- output (of the correlational structures found to be present in the sentence).

However, where the program is concerned, the different subroutines representing these sub-procedures interact continually and cannot be neatly separated. To follow the description of the program, it will be essential to keep in mind the General Description of the Procedure (pp. 6-11).

Input

3.10 The input sentence is composed of punched cards type 40, representing the words of the sentence (cf. 2.11 ff), and types 50, 51, and 60, representing punctuation marks (cf. 2.20-2.22).

3.11 The machine reads the whole sequence of input cards and records the data supplied by them in a work area (Sentence Store). During this operation the words are given their Input Numbers, i.e. sequential numbers from 1 to 16, which reflect the word-order of the input sentence.

3.12 In the record of the first word the first S-number (cf. 1.15) is then read, and the data corresponding to that item are transferred to the Multistore area.

3.13 Each Ic (cf. 1.21) consists of 6 characters:

4 characters,	xxxx	=	correlator number;
1 character,	y	=	correlator type;
1 character,	z	=	correlational function.

The character y determines the specific section of a transformation table which indicates the operational code of the Ic-number characterised by y.

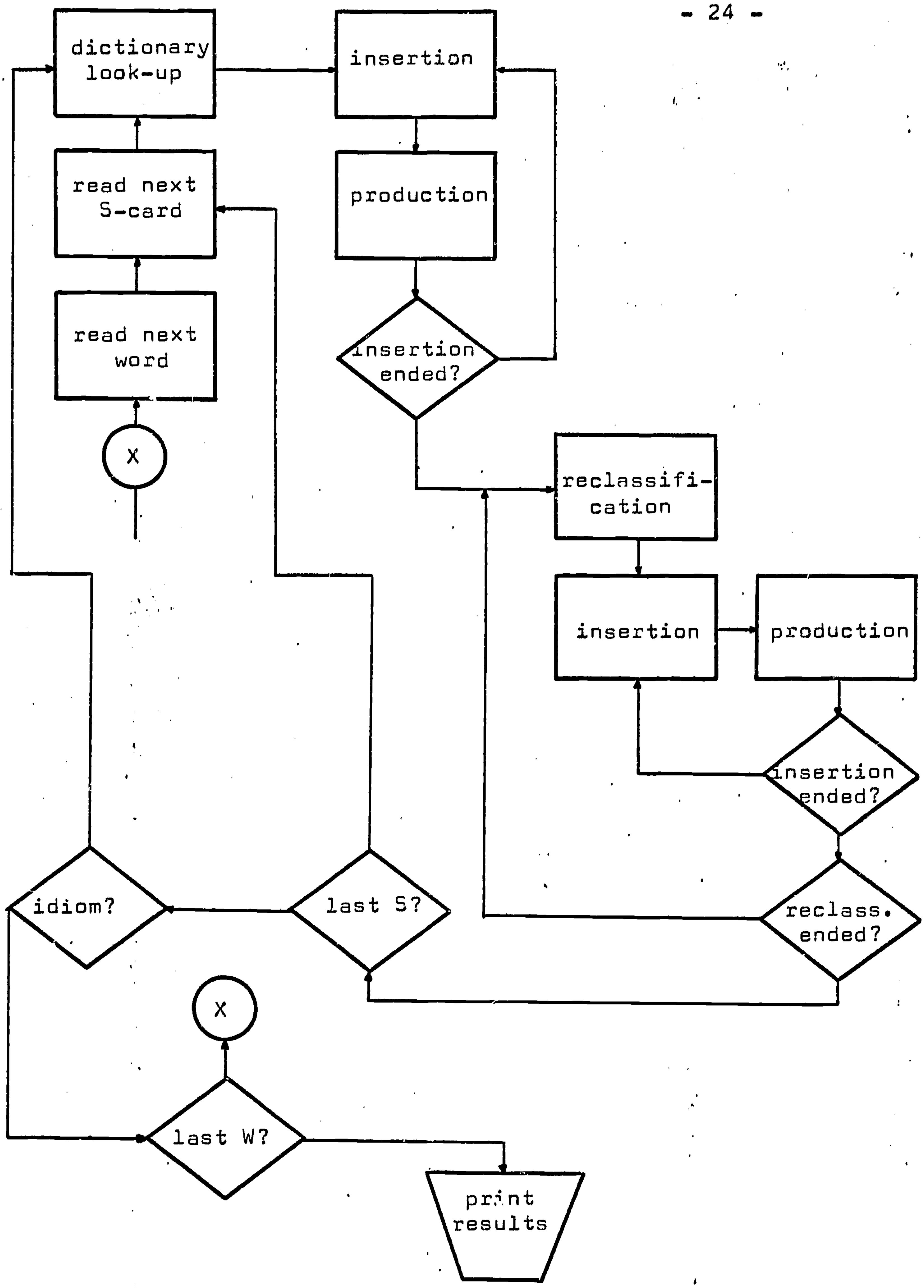

dictionary look-up
insertion
read next S-card
production
read next word
insertion ended?
X
reclassification
insertion
production
insertion ended?
idiom?
last S?
reclass. ended?
X
last W?
print results

Fig. 2

3.14 The transformation table is divided into 7 sections; six of them correspond to the different combinatorial 'modes' of the six correlator types (cf. 1.23); the seventh section is for <u>Recognition Indices</u> Ri (cf. 6.22), which play no part in the making of correlations but serve to specify items for the reclassification procedure.

The different sections of the table thus reflect the division into sections of the Multistore area; that is to say, those sections of Multistore columns that represent Ic's (the 8th and left-most Multistore section (cf. 3.17) needs no counterpart in the transformation table).

3.15 The table contains a total of 496 Ic-numbers divided into correlator types as follows:

E:	119	M:	62
F:	93	N:	183
G:	2	V:	7
		Ri:	30

3.16 The characters xxxx are transformed, by means of the transformation table, into <u>relative addresses</u> of the form aaa, ranging from 032 through 527, each of which corresponds to the specific Ic on the lines of the Multistore area, each line extending from position 000 to position 527.

3.17 The Multistore area comprises 330 lines which occupy a total of 174,240 bytes. Of these bytes 163,680 are directly addressable by means of Ic-numbers, whereas 10,560 bytes (i.e. 32 bytes in every line) are not, because they contain the processing definitions of the items represented by the lines (left section of the Multistore area). (<u>See Fig.3</u>, page 26.)

3.18 The character z, representing the correlational function CF of the Ic, determines the configuration of bits 0, 1, 2, 3, 4, within the byte addressed by the Ic.

Multistore Area

left section,

head of lines

(words + input N^os, or P's + their specification)

7 column-sections,

6 sections for Ic's representing the 6 correlator types,

1 section for recogn. indices.

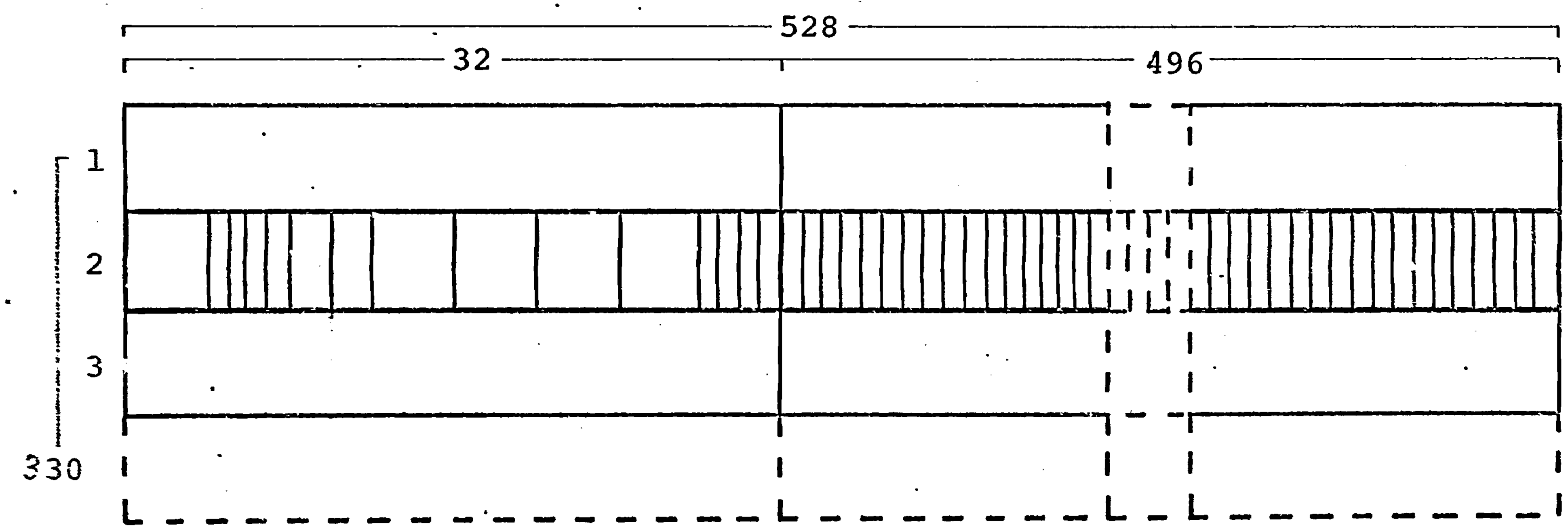


Fig. 3

The vertical numbers (1, 2, 3 .... 330) indicate <u>Multistore lines</u>, which are one byte high.

The 7 right-hand sections consist of 496 columns, which are one byte wide.

The total area covered is 174,240 bytes.

Bit 0	set	ON	represents	CF4
" 1	"	"	"	CF3
" 2	"	"	"	CF2
" 3	"	"	"	CF1

3.19 Bit 4 is set ON whenever one of the bits 0, 1, 2, or 3 is set ON. This additional bit can, subsequently, be set OFF to block access to bits 0, 1, 2, 3 of the specific byte. - This is one realisation of the 'blocks' that implement correlation restraints (cf. 7.11); in this case an individual Ic-marker of an item can be temporarily excluded from either combination or reclassification routines during the analysis of a sentence.

Fig. 4

bit N°:		
0	---	CF4
1	---	CF3
2	---	CF2
3	---	CF1
4	---	'block' (access to bits 0, 1, 2, 3
5	---	reclassification coding (cf. (6.00 ff.)
6	---	
7	---	

3.20 The transformation table (cf. 3.13-3.19) thus serves to prepare the data taken from the word entry in the vocabulary for insertion into the operational Multistore.

Insertion follows the sequence of the words and their S-numbers in the sentence store.

3.21 The first line of the Multistore contains the data corresponding to the first S-number (sense) of the first word; the second line those corresponding to the second S-number of that word, and so on for all its discriminated senses.

3.22 Correlations can connect only different words (or senses of different words), but not different senses of one and the same word. The correlation procedure, therefore, begins only when the data representing the first S-number of the second word are being inserted.

From this point on, correlations may result from the procedure. They are called Products, and each one of them occupies one line of the Multistore area; the sequence of occupation, therefore, will be determined by the results of the correlation procedure as well as by the sequence of words in the input sentence.

3.23 The first lines of the Multistore are always occupied by the S-items of the first word; they constitute 'Level' number 1. Then comes the first S-item of the second word (beginning of level number 2); if this produces a correlation with one of the S-items of the first word, the resulting product will occupy the next line of the Multistore. The lines containing the S-items of the second word, therefore, may not be consecutive because there may be product lines between them.

3.30 When the products arising from the last S-item of a word have occupied their lines, the level of that word is closed and the next level begins with the insertion of the first S-item of the next word in the input.

Correlation Procedure

4.00 The making of a correlation is subordinated to three conditions:

a) the items to be combined must be contiguous;
b) the two markers representing the two items respectively in one and the same Ic-column must be complementary;
c) there must be no block affecting that correlation.

4.01 In order to gain both space and speed, it was desirable to keep the correlation routine as homogeneous as possible for all types of correlator. For this reason the correlation routine works in the same direction regardless of the word-order of correlata in the sentence; it works from right to left, whether the construction be normal or inverted.

To distinguish between the operational order and the syntactical order, we speak of 'right-hand' pieces and left-hand pieces when we refer to the correlation routine - and it should be remembered that these terms do not necessarily correspond to 'first correlatum' and 'second correlatum' respectively.

e.g. In the phrase: "John must", the word "John" is both 1st correlatum (CF1) and 'left-hand' piece of the correlation; in the inverted form of the same correlation, "must John", however, "John" is still the 1st correlatum, but now it has the position of 'right-hand' piece.

4.02 In the correlation procedure all attempts to combine one item with another start from a right-hand piece (RH), i.e. from the later item in the word order of the sentence. Thus, whenever an RH piece is inserted, the preceding part of the specific Ic-column is searched for a left-hand (LH) piece. If no LH is found, no correlation can be made in that column. If a LH is found, other checks have to be made to ascertain that a correlation is possible.

(Note: here and in the following, 'right-hand piece' or RH refers to a word or word combination as represented by those of its Ic's which indicate correlational possibilities with the word or word combination on its left; and LH refers to a word or word combination as represented by those of its Ic's which indicate correlational possibilities with the word or word combination on its right. Any word or word combination, therefore, can be both RH and LH, depending on which of its Ic-strings one is considering at the moment.)

4.04 The first check (cf. 4.00,a) concerns the contiguity of the two items represented by RH and LH respectively.

Two items are contiguous if the right-most word of the left-hand item is the the word immediately preceding the left-most word of the right-hand item.

e.g. in "John works" the two words are contiguous;
in "John works hard" the word "John" and the product "works hard" are contiguous;
in "John works hard to make a decent living" the product "works hard" is contiguous with the product "to make a decent living", and "John" is contiguous with the product resulting from these two.

Contiguity is checked by means of the 'Level-Indication' assigned to each word during input (input number) and to each product during production (cf. samples of print-out, Appendix I-a).

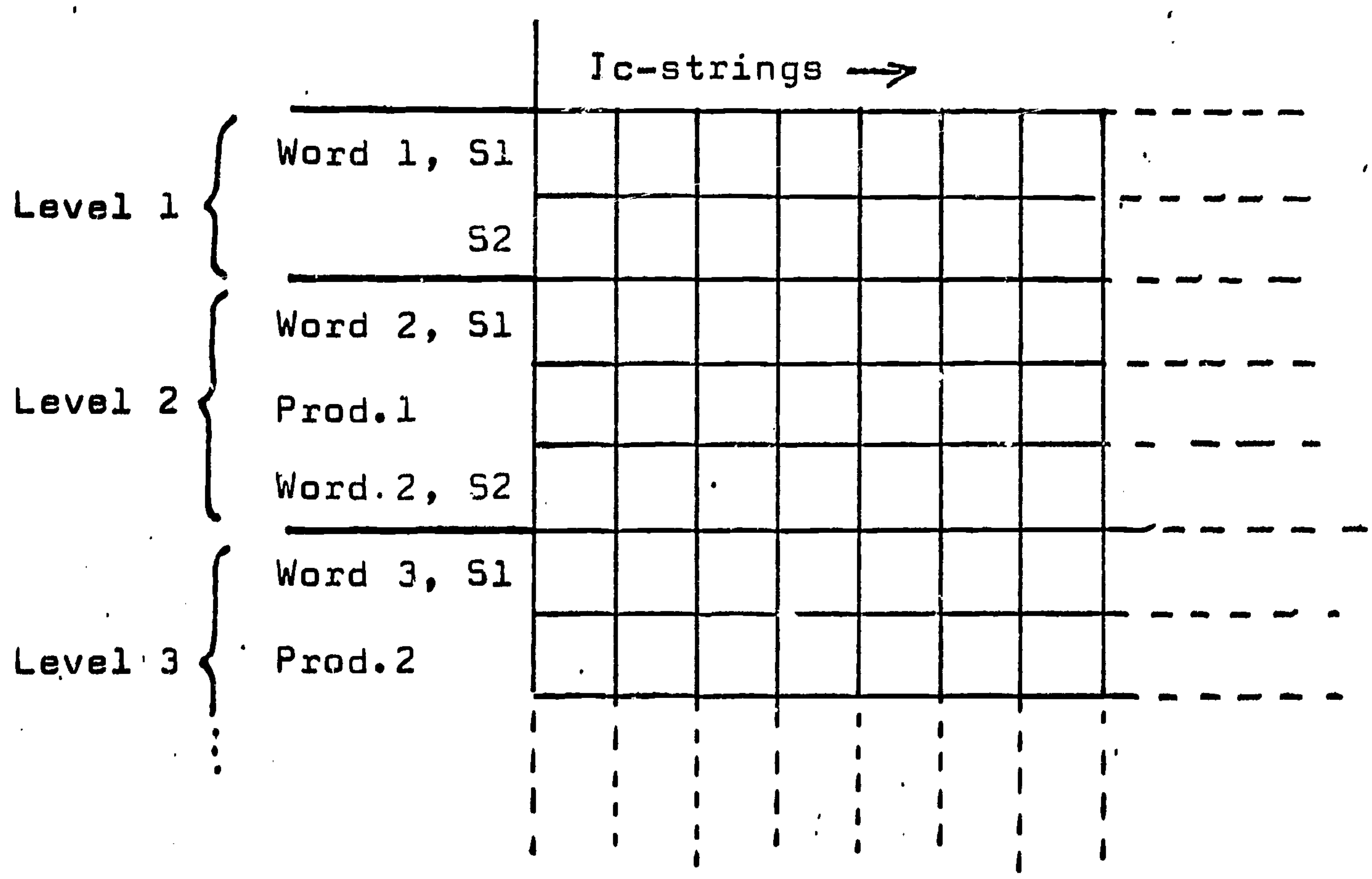


Fig. 5

4.10 The second check (cf.4.00,b) concerns the compatibility of RH and LH. This compatibility is determined by the sequence of correlational functions required by each correlator type.

The required sequences are:

Correlator type	E:	E1	+	E3	+	E2	,
" "	G:	G1	+	G3	+	G2	,
" "	F:	F3	+	F2	+	F1	,
" "	M:		M2	+	M1 ,		
" "	N:		N1	+	N2 ,		
" "	V:		V1	+	V2 .		

4.11 Correlator types E, G, and F have three elements (since the correlator is here explicit, i.e. it is expressed by a word of the sentence).

In order to keep the correlation procedure as homogeneous as possible, these explicit correlations are made in two steps: the first step correlates two of the three CF's to form a Semiproduct, and the second step then correlates this semiproduct with the third CF.

(Besides homogenising the procedure, this makes it possible to let certain explicit correlators form adverbial or modal phrases, if this should be desired; e.g. "he arrived in a car" should form an explicit correlation, whereas in "he arrived in a rage" the semiproduct "in a rage" should be treated as adverbial to "arrived". This second possibility has been implemented in the system; it has not yet been used because further text analysis will be required to establish its exact range of application.)

4.12 Semiproducts, thus, can be reclassified in two different ways:

1) as parts of a three-piece explicit correlation - and in this role they receive correlational function CF4 of the Ic that was responsible for their formation; this CF4 is assigned to them by an automatic rule.
2) as ordinary correlations (modal, adverbial, etc., phrases), in which role they are assigned the required Ic's by ordinary reclassification Rules, in the same way as any other product of an implicit correlator.

In all other respects semiproducts (SP's) are treated like other items, i.e. like words or ordinary products.

4.13 The correlation of explicit products is achieved in the following sequences:

```
Correlators type E:        E1_________E3
                           ----,-----
                               SP,E4_______________E2
                                  -------,-------
                                        P(E)

Correlators type F:        F3_________F2
                           ----,-----
                               SP,F4_______________F1
                                  -------,-------
                                        P(F)

Correlators type G:                  G3_________G2
                                     ----,-----
                           G1_______________SP,G4
                           -------,--------------
                                 P(G)
```

4.20 The <u>correlation procedure</u> is divided into three types of mechanism, or <u>Modes</u>, according to the roles the right-hand and left-hand pieces play in it.

4.21 <u>Mode 1</u>, i.e. the passive role played by LH-items (cf. 4.02). All Ic's representing LH-items are inserted as markers into their specific columns of the Multistore and remain passive, as targets, as it were, for the searches triggered by RH-items.

This mode applies to Ic's of the following correlator types and CF's:

E1, G1, N1, V1; M2; F3, G3; E4, F4.

4.22 <u>Mode 2</u>, i.e. the active role played by RH-items in the production of semiproducts. Ic's representing possible RH-items of semiproducts are inserted as markers and trigger a search (in the column into which they are inserted) for markers of complementary LH-items.

Mode 2 applies to Ic's of the following correlator types and CF's:

E3 (which combines with E1),
F2 (which combines with F3),
G2 (which combines with G3).

4.23 Mode 3, i.e. the active role played by RH-items in the production of full products (correlations). Ic's representing possible RH-items of full products are inserted as markers into their specific columns and trigger a search (in that same column) for markers of complementary LH-items.

This mode applies to Ic's of the following correlator types and CF's:

E2 (which combines with E4),
F1 (which combines with F4),
G4 (which combines with G1),
M1 (which combines with M2),
N2 (which combines with N1),
V2 (which combines with V1).

4.30 The third check (cf. 4.00,c) concerns 'blocks' which may or may not have been recorded for a specific S-item or product during the analysis of a given input sentence.

A block prevents the item from becoming a left-hand correlatum in the subsequent course of the analysis procedure. There are two forms of this kind of block:

a) temporary, which can be removed at a subsequent point of the analysis;

b) permanent, which remains operative for the duration of the sentence analysis in course.

4.31 The temporary block is indicated by bit 2, the permanent block by bit 3 of byte 5 (see Fig.4,p.27); an example of such a block is given under 7.21.

Production

5.00 If the three checks (cf. 4.00) turn out positive, a product is generated.

Since the characteristics of the items that form the product are implicit in the locations (addresses) of these items within the Multistore area, "generating" a product means, in fact, no more than inserting the relevant characteristics of its components at the head (i.e. the left-most part) of the next free Multistore line.

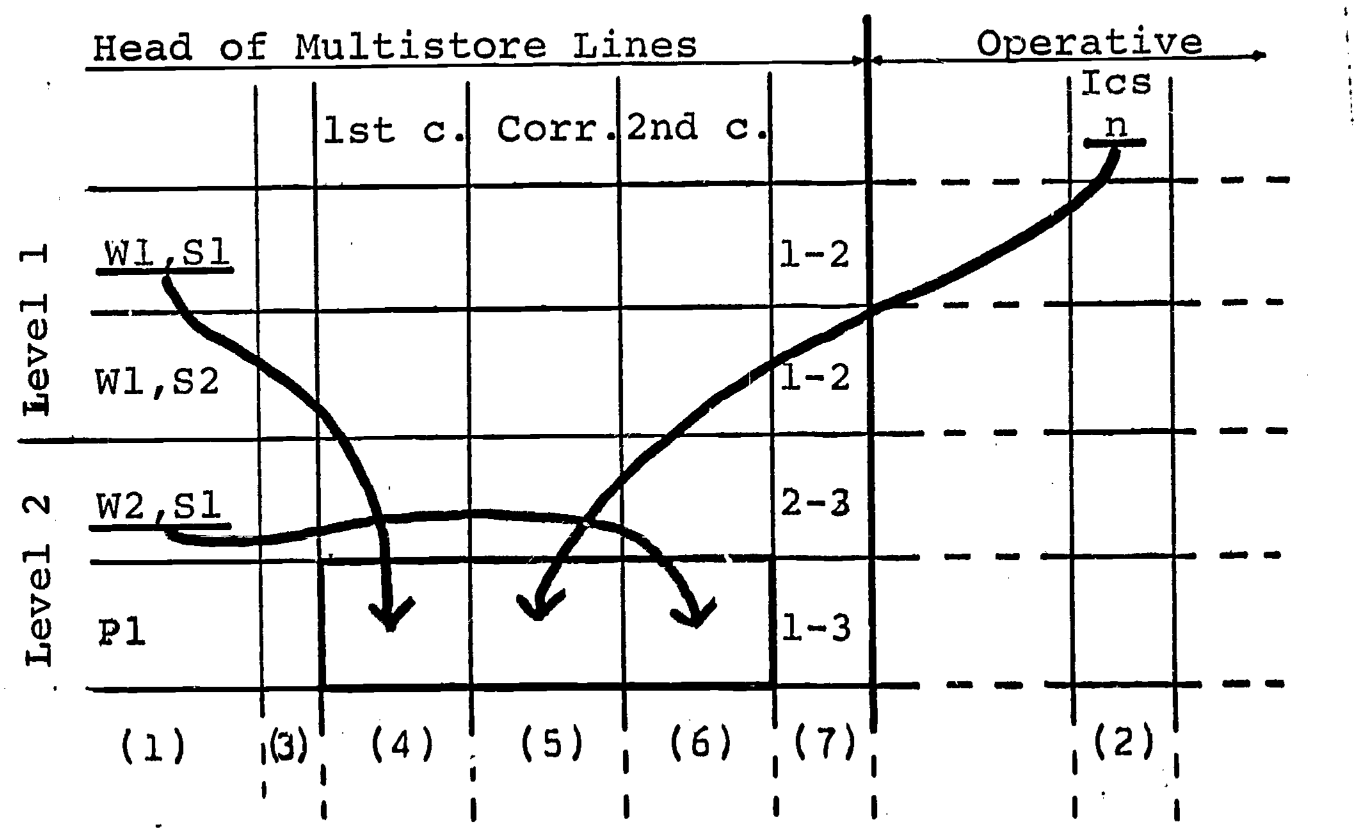


Fig. 6

5.01 The data recorded at the head of the Multistore line are:

1) Serial number of product or semiproduct;
2) Ic-column, origin of product;
3) G & T code of product;
   (this code is, at present, used for single words only);
4) 1st correlatum of the product (i.e. its address);
5) Correlator Nº and type (i.e. a number representing the increment required to address the Multistore column dedicated to that correlator);

6) 2nd correlatum of the product (i.e. its address);

7) Level range of the product (i.e. the lower level-number of its first component and the higher level-number of its last component).

5.02 Production takes place as long as new Ic's are inserted on the given level - provided, of course, that they find complementary markers in their column and that the relevant checks are positive).

5.03 The reclassification of newly-made products takes place when there are no more Ic's to be inserted on the given level.

Reclassification

6.00 The reclassification routine begins by reclassifying the first product recorded on the present level.

6.01 The data that govern reclassification are contained in the Lists and Rules (cf. 1.40-1.54).

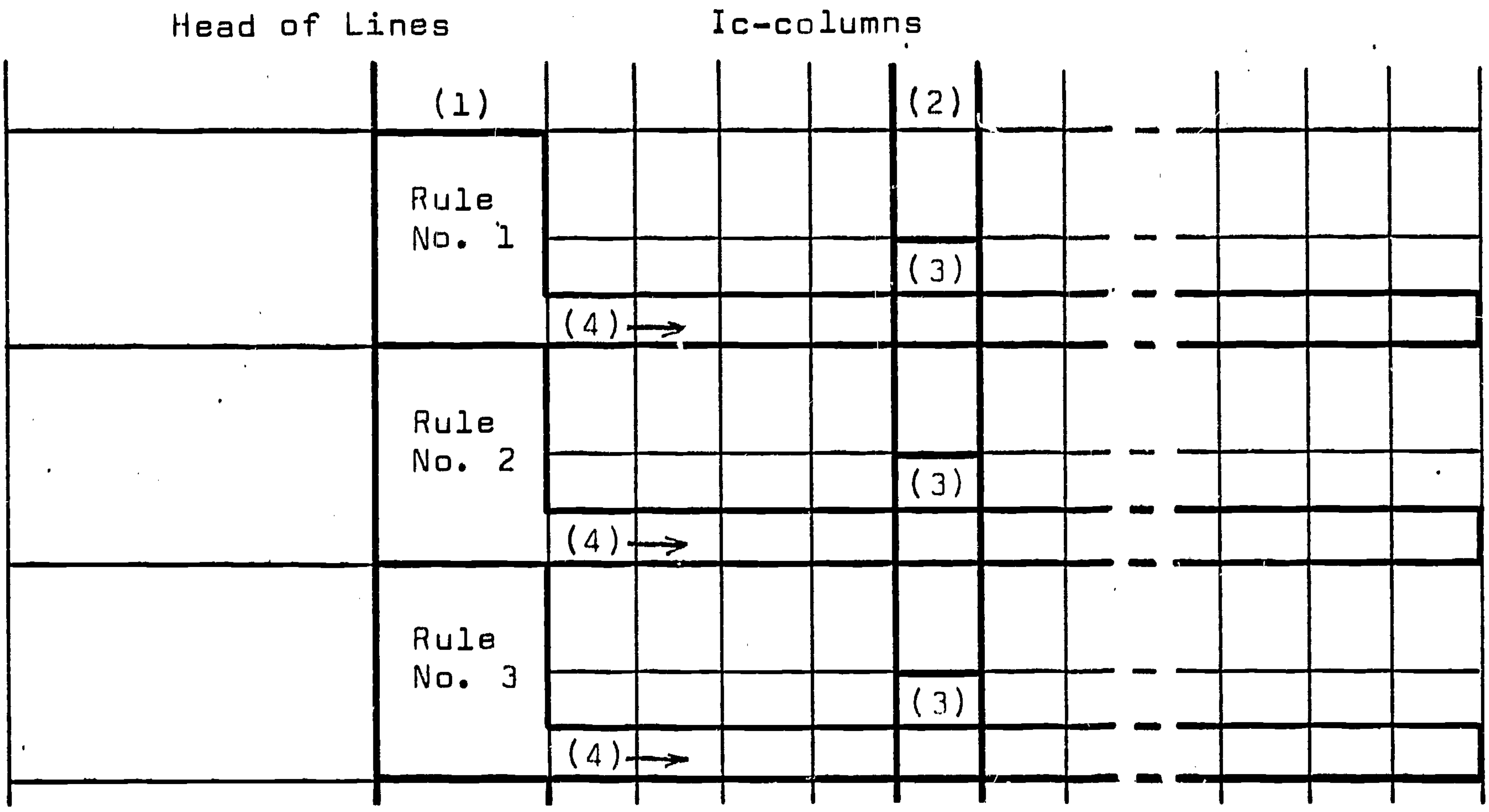


Fig. 7: (1) Rule specifications; (2) Column of producing Ic; (3) Rule-markers constituting the Ic's List; (4) Ic-markers constituting the string of the Rule's assignations.

6.02 The reclassification List for the products made by a specific correlator is incorporated in the Multistore column that represents that correlator, and the reclassification Rules are incorporated in the Multistore lines.

6.03 Since each List contains the code numbers of the Rules that are to be applied to the products of the correlator to which it refers, the List's representation in the specific Multistore column consists of Rule-markers in all those bytes that are intersections of that column with the lines containing the listed Rules. (see Fig.7, page 35).

6.04 In reclassification the Multistore column is scanned from the top down. When the first Rule-marker is encountered, the scanning shifts to the head of the line in which the marker was found; this left-most section of the line (positions 28-31) contains the specifications of the first reclassification Rule applicable to the product that is being reclassified.

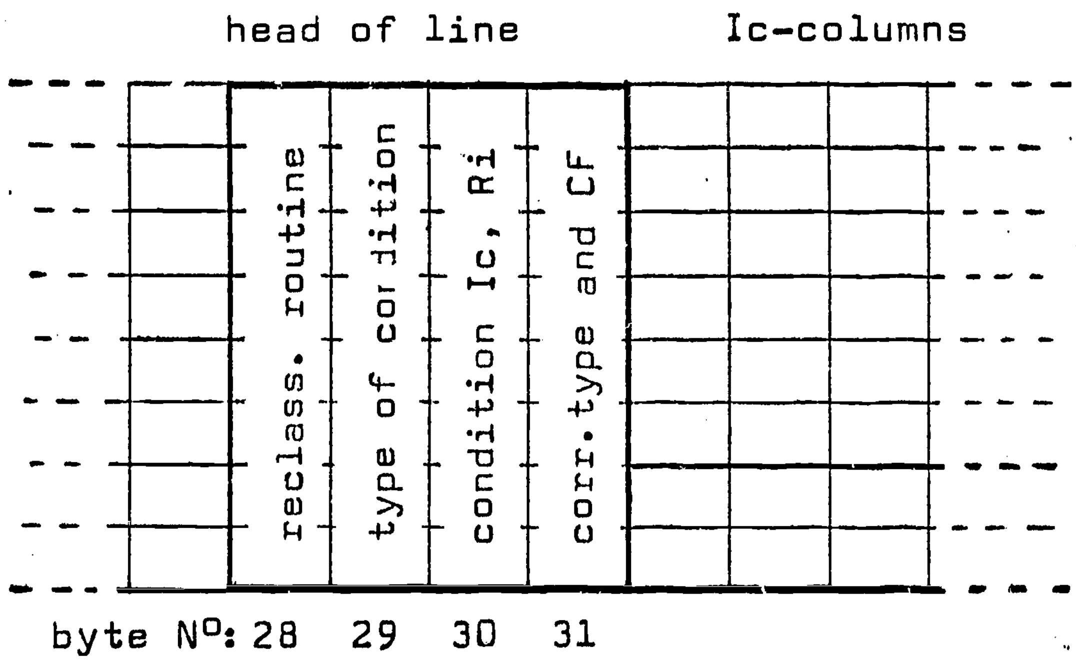


Fig. 8

6.10 Byte 28 of the Rule specification contains the incrementation value by means of which the address is computed at which the modality (i.e. the application routine) for the particular Rule is stored.

Byte 29 is made up in the following way:

6.11 Bit 6 - if set ON - blocks the Rule represented by the line; this block is used for experimental purposes (for instance if one wants to process a sentence both with and without a specific reclassification Rule). The Rule block is set up and removed during the analysis by internal commands triggered by semantic or other factors.

6.12 Bit 7 discriminates those Rules which apply to semiproducts only.

6.13 Bits 0, 1, 2, 3 indicate the type of Rule represented by the Line in question.

6.20 Reclassification Rules fall into two groups:

Rules that assign Ic's to a given product unconditionally, and
Rules that assign Ic's only if the product that is to be reclassified satisfies a specific condition.

Actual codification of Rule types:

1100 rules that specify a string of Ic's (to be assigned unconditionally or conditionally);

0100 rules that transfer Ic's from one of the product's correlata to the product (with or without exceptions);

0011 special rules to trigger specific 'blocks';

0001 special rules for experimentation with data that do not usually enter the reclassification cycle.

6.21 Unconditional Rules assign a string of one or more Ic's to the product (P) to be reclassified.

6.22 Conditional Rules are of four types:

1) Rules which assign one or more Ic's only if the 1st or 2nd correlatum (as specified by the Rule) of the

product bears a given Ic, Recognition index, or G-code. This condition is given by a character in byte 30. (These Rules can also be negative, in which case assignation is made only if the specified correlatum does not bear a specified Ic, Ri, or G-code.)

2) Rules which assign one or more Ic's only if those Ic's are part of the string of the 1st or 2nd correlatum (as specified by the Rule) of the P.

3) Rules which assign to the P all Ic's found in the relevant string of the 1st or 2nd correlatum (as indicated by the Rule) of the P.

4) Rules which assign all Ic's found in the relevant string of the 1st or 2nd correlatum (as specified by the Rule) with the exception of a set of one or more specified Ic's. (Note that Rules of type 3 are, in fact, Rules of type 4 with a zero exception set.)

6.23 Any Rule assigns Ic's of only one specified correlator type and CF, both of which indications are given in byte 31. Rules of type 2, 3, and 4, transfer Ic's from a correlatum of the P to the P itself without changing their correlator type or CF; in Rules of type 1 the correlator type and CF of the condition-Ic has no bearing on correlator type or CF of the Ic's assignable to the P.

6.30 In the remaining 496 bytes of the Multistore line - each of which bytes constitutes the intersection with one Ic-column of the Multistore - the bits 6 and 7 (cf. Fig 7, p.35) indicate whether the Ic represented by the particular column is assigned by the Rule or not.

Thus, scanning the bytes of the line, each bit 6 found set ON implicitly determines one Ic-number (the one represented by the column to which the byte belongs) that is to be assigned to the product. - The correlational function of this Ic-number, on the other hand, has already been determined by byte 31 of the line that is being scanned.

6.31 Once an Ic to be assigned by the Rule has been spotted (bit 6 set ON), it immediately effects the insertion of an equivalent Ic-marker in the byte that constitutes the intersection of that same Ic-column with the line representing the P which is being reclassified.

Since both the spotting of the Ic-marker contained in the Rule and the insertion of the equivalent Ic-marker in the Ic-string of the P take place in one and the same column (which is the column dedicated to that specific Ic), the Ic-number itself does not have to be read, recorded, or shifted: it remains implicit as locational characteristic of the specific column during the entire sequence of operations.

6.32 The end of a reclassification Rule, i.e. the last Ic to be assigned by it, is indicated by bit 7 (in the byte containing the last Ic-marker); if bit 7 is set ON, it constitutes the end signal of the particular Rule.

6.40 Reclassification can be summarised as follows:

Whenever a product is made in a column X of the Multistore, the characteristics of this product are recorded at the head of the next free Multistore line; production then continues until there are no more Ic's to be inserted on that level.

Reclassification then begins with the first of the newly recorded products.

The correlator number of the product is also the number (address) of the column that contains the reclassification List relevant to that product.

The column is then scanned for Rule-markers.

The lines on which Rule-markers are found contain the relevant reclassification Rules. At the head of the line the conditions of Ic-assignation are specified; the rest of the line contains markers indicating the specific Ic's to be assigned to the product.

Whenever a marker is encountered it effects the insertion of a marker of the same Ic in the Ic-string of the product.

This insertion is in every way equivalent to the insertion of an Ic-marker originating from an Ic of an input word.

6.41 As in the case of Ic-markers originating from the Ic-string of an input word, this Ic-marker resulting from the reclassification of a product may cause a new product to be made (cf. 4.00 ff); if it does, the new product is recorded at once at the head of the next free Multistore line (i.e. it is recorded before the reclassification of the original product proceeds).

6.50 Reclassification procedure is, therefore, continually interlinked with the correlation procedure (cf. General Description, pp. 6-11). The reason for this is the considerable gain in processing time that was made possible by the systematic superposition of two essentially different procedures - i.e. correlation and reclassification - in one and the same area of significatively structured machine memory.

Restraints (*)

7.00 Reclassification, by assigning Ic's to products, determines the possibilities these products, or word combinations, have of combining with other words. In doing this, it characterises the products in much the same way as the assignation of Ic's characterises single words. There is, however, one important difference.

* We speak of 'Restraint' when the specific context of a product makes it impossible for that product to correlate with another item in a way that would be possible and correct in other sentences.

7.01 When we assign Ic's to a word as vocabulary item, we necessarily have to consider all correlations into which the word can possibly enter; but when the procedure re-classifies a product, i.e. a word combination found to be feasible with words of the given sentence, all that has to be considered are the correlational possibilities this product has <u>within the given sentence</u>. Experience has shown that there are many cases where a certain correlational possibility of a product, although legitimate in theory, can be excluded because of what comes before the product (i.e. stands on the left of it) in the input sentence.

7.10 A simple example is this:

In the sentence "the wine is sour", the product "wine is sour" does not have to be correlated because the product "the wine" necessarily supersedes the correlational possibilities the word "wine" has by itself.

7.11 Restraints of this kind are implemented by 'blocking' the relevant Ic-markers of the item. That is to say, in the example, once the correlation "the wine" has been produced, certain Ic's of the word "wine" are prevented from forming correlations to the right (cf. 3.19).

7.20 A similar type of restraint can be formulated for certain S-items (senses) of a word which, during the analysis of a given sentence, fits as 2nd correlatum into one of several specific correlations.

7.21 An example is the sentence "my answers were complete": When the word "answers" can be correlated as a plural noun with the possessive "my", it is impossible for the verb-sense of "answers" (3rd person singular, present indicative) to be correlated within the same sentence; this other sense of the word can, therefore, be eliminated from the analysis, and this is achieved by blocking the Ic-markers in the Multi-store line containing the relevant S-card (cf. 4.30 ff).

7.30 Another type of restraint concerns 'complete' products (i.e. products that contain all the words of a sentence) which, although grammatically correct, are not acceptable as interpretations of the sentence. Such products occur when the string of words that constitutes the input sentence would give rise to a different syntactic interpretation if it were preceded or followed by something else.

7.31 A case in point are sentences containing the form "were" which can be either indicative or subjunctive and must, therefore, give rise to two correlational interpretations.

If a string such as "they were leaving" is the whole sentence, however, only the indicative interpretation is acceptable, since the subjunctive one would require "if" or "I wish" or some such conditional expression to precede it.

Thus, although the conjunctive construction is a correct interpretation as far as it goes, we can formulate the rule that it can be excluded as a 'non-sentence' if it stands alone (cf. print-out, Appendix I-d).

7.40 The last type of restraint implemented in the present version of the Multistore system is similar to the 'non-sentence' exclusion, except that it affects phrases (products) before the final level. It is applied in cases that are akin to the kind described under 7.21, but cannot be dealt with by a block because the spurious product is made before the desired alternative correlation has turned up in the procedure, i.e. before a choice has become possible.

7.41 In a sentence such as "they accepted his apology", for instance, it is inevitable (because necessary if the sentence ended with "his") that the pronominal sense of "his" produces a correlation in which it is the object of the verb; and this correlation can be discarded <u>only at a later stage</u>, i.e. when the next word has entered into the procedure. However, once the next word has entered and has, in fact, given rise to a correlation with the possessive sense of "his", the preceding verb-object correlation is definitively superseded and can be discarded. The program does this by blocking the reclassification of the superseded product, impeding thus its entering into further correlations; this retroactive block is shown in the print-out by the word DISCARD.

7.50 The capability of blocking certain items or certain Ic's has not yet been fully exploited, only the most obvious exclusions have been formulated as rules. A large-scale examination of texts would certainly bring to light many more instances of possible blocks; but, given the size of our group (one and a half linguists), we are, unfortunately, in no position to undertake large-scale text surveys. Our main object in including at least some of these rules in the program, was to demonstrate that it can handle restraints of this kind and that they go a long way to reduce the production of spurious correlations.

D - Output

8.00 The output of the present version of the parser consists of three parts:

a) a graphic display of the correlational structures that constitute the parsing of the input sentence;

b) a print-out of the intermediary products that led to the parsing;

c) a print-out of the Ic-strings assigned to the intermediary products by reclassification.

Parts b and c are invaluable for checking the functioning of the system and its grammar, but would, of course, be cut out if the system were applied to text-analysis.

8.01 Print-out begins when the correlation procedure for one input sentence has come to its end. The end of the correlation procedure is recognised by the fact that, after the operations of a given level, a full-stop card (cf. 2.21) enters instead of a card representing a next word. (Note that points belonging to abbreviations would not be represented by a full-stop card, but would be part of the relevant vocabulary item.)

8.02 The first item printed out is the string of words constituting the input sentence, showing the input numbers of the words and the punctuation marks.

e.g. 01 WE 02 WANTED 03 TO . 04 GO .

The word-string appears at the top of every page of print-outs a and b .

8.10 The print-out routine then scans, from top to bottom, the left-most section of the Multistore area, i.e. the heads of the lines, taking into account the product lines only (products and semiproducts).

8.11 When a product is found, it is examined for its level-range and printed as one line of print-out b. The following data appear in print (cf. Appendix I-a):

Product number: P for product, S for semiproduct; and four digits for the P-number.

First correlatum: W for word, P for product;
four digits for input- or P-number respectively;
two digits for the relevant S-number of the word;
two digits for the G-code of the W.

Correlator: four digits for the correlator number;
one digit for the correlator type.

Second correlatum: data as for 1st correlatum.

Level-range: input number of first word of the P;
input N° + 1 of second word of the P.

8.12 If a product begins at level 01 and ends with a full stop, the word COMPLETE is added to the line in print-out b.

8.13 If the product has been blocked by the type of restraint described under 7.41, the word DISCARD is added to the line in print-out b.

8.14 If the product has been recognised as not acceptable (cf. 7.40 ff), the words NON-SENTENCE are added to the line in print-out b.

8.15 If a 'complete' product is affected by neither of these restraints (cf. 8.13, 8.14) print-out a takes place.

8.20 The display of a 'complete' product's correlational structure (print-out a) is composed by retracing, in the product records, first all the second correlata involved in the product and then all the first correlata; the graphic arrangement is constructed line by line.

(A complete explanation of the data in print-outs type a is given in Appendix I-a.)

8.30 When print-outs a and b have been completed, the product records are scanned once more and, whenever a product or semiproduct is encountered, the corresponding Multistore line is scanned for Ic-markers (reclassification).

8.31 Each Ic-marker found is identified as to correlator number, type and correlational function, and the full Ic's are printed consecutively, thus displaying the reclassification string of the product in hand (cf. Appendix I-a).

Bibliography

(of relevant previous reports)

(1) -- First Draft of an English Input Procedure (E.v. Glasersfeld and Jehane B.Burns), TSR No.8, Contract AF61(052)-362; Methodos, vol.XIV, No. 54, Milan (Italy), 1962.

(2) T-3 A Project for Automatic Sentence Analysis (E.v. Glasersfeld), Beiträge zur Sprachkunde und Informationsverarbeitung No.4, Munich (Germany), 1965.

(3) T-10 Multistore: A Procedure for Correlational Analysis (E.v.Glasersfeld, P.P.Pisani, J.B.Burns), Automazione e Automatismi, vol.IX, No.2, Milan (Italy), 1965.

(4) T-11 Automatic English Sentence Analysis (E.v.Glasersfeld and P.Terzi), Final Report, Grant AF EOAR 64-54, IDAMI Language Research Section, Milan (Italy), 1965.

(5) T-12 An Approach to the Semantics of Prepositions (E.v.Glasersfeld), Symposium on Computer-Related Semantic Analysis, Las Vegas (Nevada); published by Wayne State University Press, 1965.

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

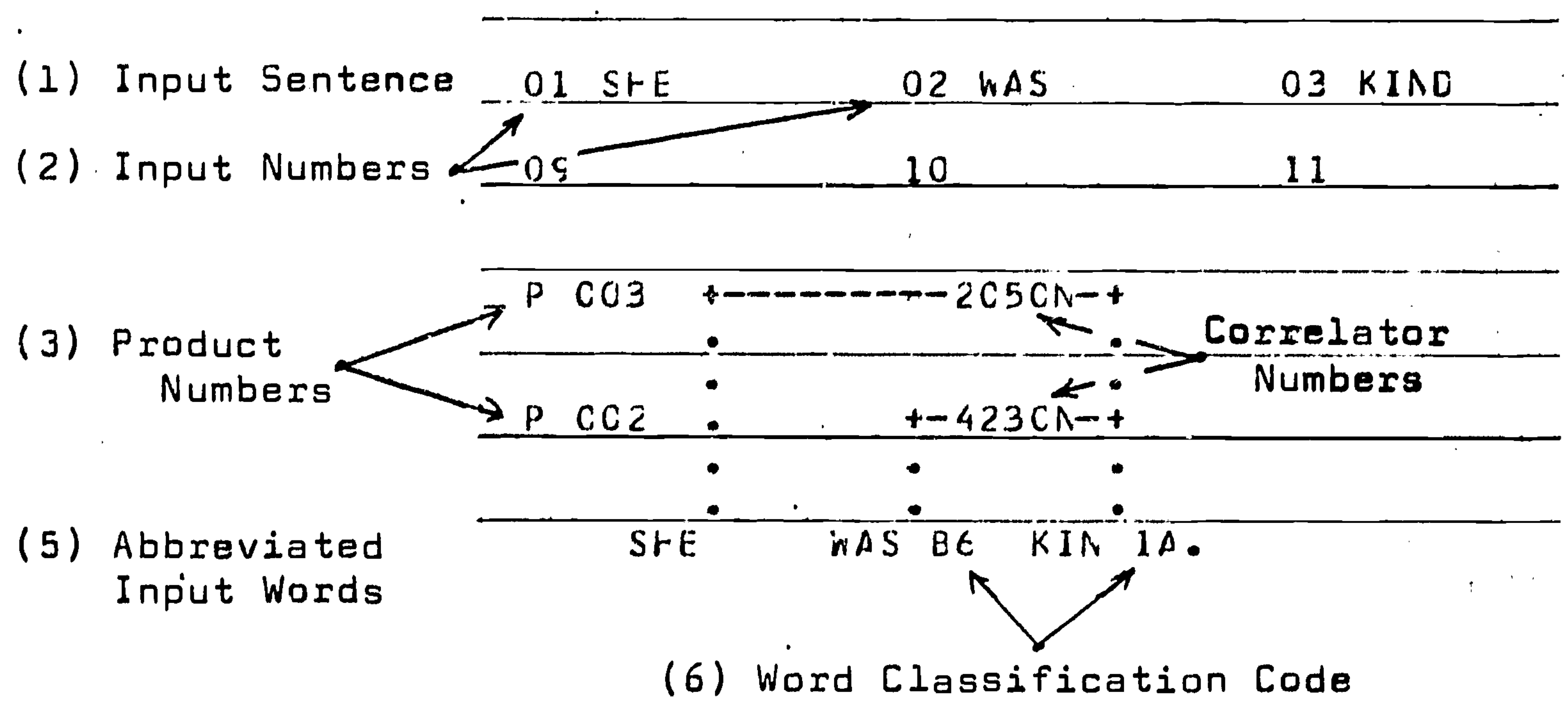
(7) CG-1 Some Adjective Classes Derived from Correlational Grammar (E.v.Glasersfeld and B.Notarmarco), Georgia Institute for Research, Athens (Georgia), 1968.

(8) CG-2 An Introduction to Correlational Grammar (Brian Dutton), Georgia Institute for Research, Athens (Georgia), 1968.

(9) -- A Scheme for Semantic Controls in Automatic Sentence Analysis (Jehane Burns), Progress Reports Nos 1 and 2, Grant AF EOAR 67-29, Literary and Linguistic Computing Centre, University of Cambridge (England), 1967-68.

A P P E N D I X I-a

Graphic Display of Sentence Structure



1) The input sentence is limited to 16 words, which have their pre-established numbered places (two lines of 8 places each).

2) The input numbers reflect the words' position in the sentence; they are printed out because in the product list (see page I-a,2) the individual words are identifiable only by their input numbers.

3) The product numbers reflect the sequence of production in the course of the analysis; the first product number, in the left top corner of the graphic display, is the number of the displayed product.

4) The correlator number specifies the correlator responsible for the correlation indicated by the dashes on either side of it; if the correlation is 'normal' the number appears at the end of the line, if it is 'inverted' the number stand at the beginning.

5) At the terminals of the structure, the three first let-

ters of the relevant word are printed.

6) the word code contains a summary of grammatical and semantic data (cf. G-code, 1.16); it serves exclusively for the identification of the particular word's sense (S-number).

Product List

(Print-out type b) Reproduction and explanation on page I-a,3

Reclassification Print-out

This print-out (type c) shows the strings of Ic's assigned to the products by the reclassification routines. It serves the checking of reclassification rules and the tracing of errors.

```
P 0001          001CE1,0010E2,0020E1,002CE2,003CE1,0030E2,0460E2,010CF1,
P 0001          2570M1,4210M1,4230M1,6330M2,6350M2,6351M2,7010M2,3010N2,

           0170F1,0240F1,0241F1,0300F1,0410F1,0460F1,0500F1,0550F1,0750F1
           3030N2,3380N2,3390N2,4550N2,6310N2,6330N2,6350N2,6351N2,7010N2


P 0002          0010E1,001CE2,0C20E1,0C2CE2,010CE1,017CE1,024CE1,03C0E1,

           0410E1,0460E1,0550E1,0750E1,4410M1,2050N2,4410N1,4420N1,


P 0C03          001CE1,0010E2,0020E1,0C2CE2,003CE1,0030E2,0460F2,0100F1,
P C003          3730M1,3731M1,6330M2,635CM2,6351M2,7010M2,3010N2,3030N2,
P COC3          6351N2,7C1CN2,      ,      ,      ,      ,      ,      ,

           0170F1,0240F1,0241F1,0300F1,0410F1,0460F1,0500F1,0550F1,0750F1
           3380N2,3390N2,3670N1,3730N1,3731N1,4550N2,6310N2,6330N2,6350N2
                ,      ,      ,      ,      ,      ,      ,      ,
```

Product List

01 SHE	02 WAS	03 KIND . 04	05	06	
09	10	11	12	13	14
P 0001	W 0001 01 $1	2050 <	W 0002 01 B6		01 03
P 0002	W 0002 01 B6	4230 <	W 0003 01 1A	.	02 04
P 0003	W 0001 01 $1	2050 <	P 0002	.	01 04 COMPLETE

- P = product, S = semiproduct
- product number
- W = word, P = product
- word- or product-number of 1st correlatum
- S-number of 1st correlatum
- G-code of 1st correlatum
- correlator number
- correlator type
- W = word, P = product
- word- or product-number of 2nd correlatum
- S-number of 2nd correlatum
- G-code of 2nd correlatum
- punctuation mark at end of P
- <u>lower level</u> of product
- <u>upper level</u> of product (input number + 1 of last word contained)

A P P E N D I X I-b

Complete Parsing

Input Sentence:

01 HE	02 WANTED	03 ME	04 TO
05 ANSWER	06 THE	07 LETTERS	08 BY
09 SUNDAY	. 10	11	12

Product List (Print-out type b):

P 0001	W 0001 01 $1	2250 N	W 0002 03 F8	01 03
P 0002	W 0002 03 F8	4010 N	W 0003 01 A1	02 04
P 0003	W 0002 01 FP	4010 N	W 0003 01 A1	02 04
P 0004	W 0001 01 $1	2250 N	P 0002	01 04
S 0005	P 0003	0750 E	W 0004 01 0/	02 05
S 0006	P 0002	0750 E	W 0004 01 0/	02 05
P 0007	W 0004 03 00	2310 N	W 0005 01 VS	04 06
P 0008	P 0003	3550 N	P 0007	02 06
P 0009	P 0002	3550 N	P 0007	02 06

P 0010	W 0001 01 $1	2250 N	P 0009	01 06
P 0011	W 0006 01 CA	5010 N	W 0007 01 %@	06 08
P 0012	P 0007	4110 N	P 0011	04 08
P 0013	P 0007	4010 N	P 0011	04 08
P 0014	P 0003	3550 N	P 0013	02 08
P 0015	P 0002	3550 N	P 0013	02 08
P 0016	W 0001 01 $1	2250 N	P 0015	01 08
S 0017	P 0013	0241 E	W 0008 01 0/	04 09
S 0018	P 0011	0241 E	W 0008 01 0/	06 09
S 0019	P 0013	0243 E	W 0008 01 0/	04 09
S 0020	P 0013	0245 E	W 0008 01 0/	04 09
S 0021	P 0011	0245 E	W 0008 01 0/	06 09
S 0022	P 0011	0249 E	W 0008 01 0/	06 09
S 0023	P 0011	0253 E	W 0008 01 0/	06 09

S 0024	P 0013	0255 E	W 0008 01 0/	04 09	
P 0025	P 0013	0243 E	W 0009 01 $3	. 04 10	
P 0026	P 0003	3550 N	P 0025	. 02 10	
P 0027	P 0002	3550 N	P 0025	. 02 10	
P 0028	W 0001 01 $1	2250 N	P 0027	. 01 10	COMPLETE

Graphic Display (Print-out type <u>a</u>):

```
P 028    +------------------------------------------------------------------------2250N-+
         .                                                                              .
P 027    .         +--------------------------------------------------------------3550N-+
         .         .                                                                    .
P 025    .         .                   +------------------------------------------0243E-+
         .         .                   .                                                .
P 013    .         .                   +---------------------4010N-+                    .
         .         .                   .                           .                    .
P 011    .         .                   .                   +-5010N-+                    .
         .         .                   .                   .       .                    .
P 007    .         .                   +-2310N-+           .       .                    .
         .         .                   .       .           .       .                    .
P 002    .         +-4010N-+           .       .           .       .                    .
         .         .       .           .       .           .       .                    .
     HE         WAN F8  ME   A1   TO   OO   ANS VS   THE CA   LET %@   BY         SUN $@.
```

Explanation of Correlators:

2250N subject // past tense of verb.

3550N framing verb* + object // infinitive;
the object of the verb is the actor of the infinitive.

0243E temporal limitation, post-terminal, expressed by "by".

4010N verb // object.

5010N definite article // noun.

2310N "to" // supine (forming infinitive).

(Note: the sentence "he wanted the machine to answer the letters by Sunday" would yield two parsings: one similar to the above, and another in which the second correlation from the top would be a product of correlator 3670N, in which the subject of the verb is the actor of the infinitive, which latter expresses the final cause or purpose of the subject's activity.)

---

* i.e. clause-governing verb.

2

A P P E N D I X   I-c

An Application of Prepositional Analysis

As an example of the way in which correlational grammar can handle the relations expressed by explicit correlators (mostly prepositions and conjunctions) a sample set of eight relations collectively represented in English by the preposition "by" was introduced into the parsing system.

The eight relations and their individual correlator numbers are:

0241 Specification of lighting
e.g. "to travel by day"
"we played by moonlight"

0243 Temporal limitation, post-terminal
e.g. "she will be gone by Sunday"

0245 Spatial proximity
e.g. "he sat by the fire"
"the house by the church"

0247 Efficient agent
e.g. "he was killed by the gangsters"

0249 Authorship
e.g. "a book by Hemingway"
"a sonata by Mozart"

0251 Itinerary
e.g. "he arrived by the fields"
"I escaped by the back door"

0253 Means of transport
e.g. "they travelled by car"

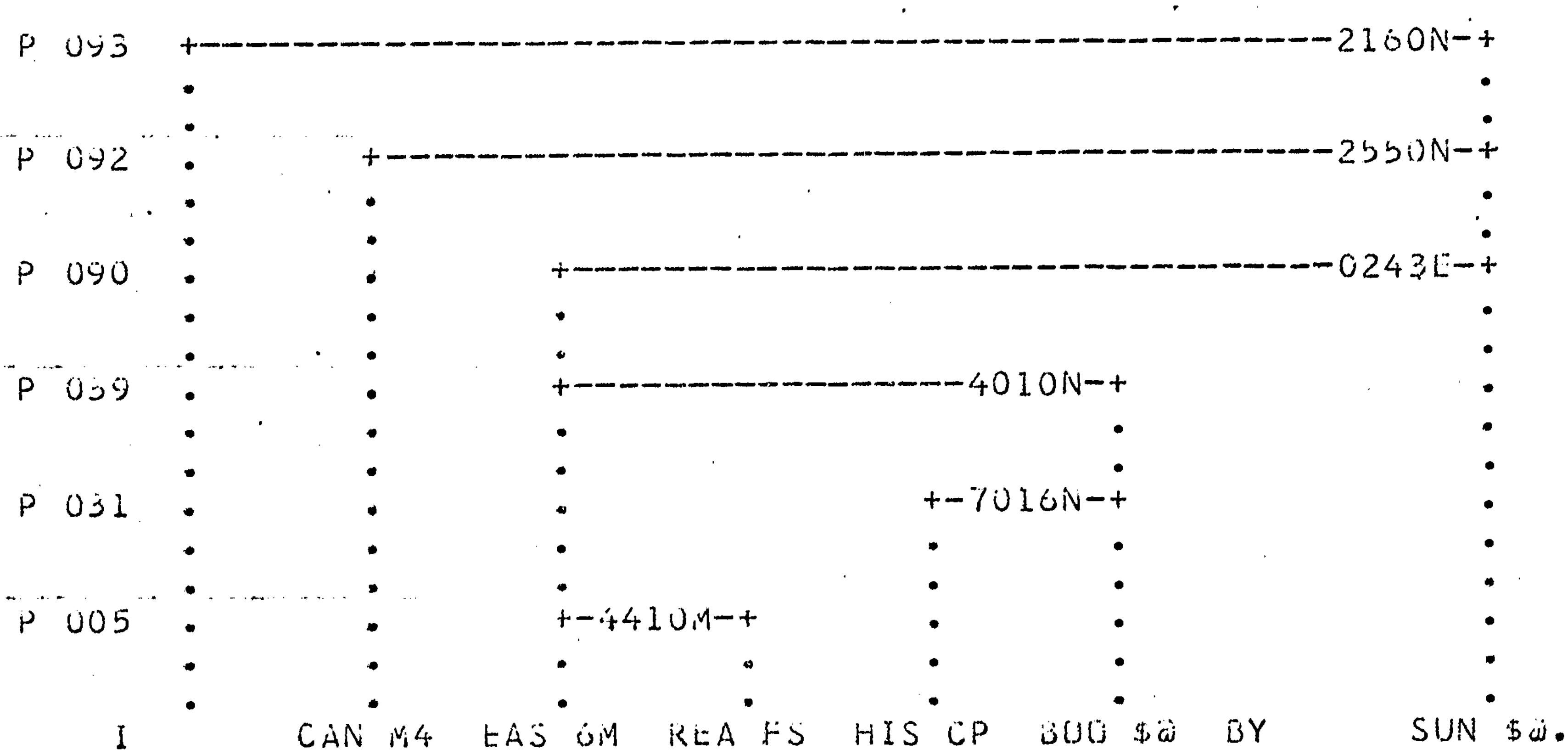
0255 Method: activity (present participle)
e.g. "he learnt it by watching professionals"

These, obviously, are not all the relations that can be expressed by the English "by" (according to a preliminary analysis, about 25 can be isolated and defined - not taking into account subdivisions based on the grammatical character of the correlata); they are, however, among the most frequent and are quite sufficient to demonstrate the system's capacity to eliminate 'pseudo-ambiguities' (cf. An Approach to the Semantics of Prepositions, 1965).

The words of the parser's vocabulary having been examined for their individual correlability within the range of the selected eight relations, the required Ic's were assigned to the vocabulary items and reclassification rules were written to assure assignation to relevant products.

The assignation of these Ic's to words of the vocabulary was done intuitively and would, of course, require verification in a large and truly representative corpus of texts before it could be considered definitive. Nevertheless, the results obtained with a tentative application are very satisfactory.

A sentence such as "I can easily read his book by Sunday" yields one interpretation only:



<u>Note</u>: Correlator 0243 is defined as '<u>Temporal Limitation</u>'

The sentence "Jones was killed by the river" yields two interpretations:

```
P C18  +-----------------------------------------2050N-+
       .                                               .
       .                                               .
P C15  .       +-------------------------------0247E-+
       .       .                                       .
       .       .                                       .
P C11  .       .                          +-5010N-+
       .       .                          .            .
       .       .                          .            .
P 002  .       +-2570N-+                  .            .
       .       .       .                  .            .
       .       .       .                  .            .
    JON     WAS BE  KIL VP  BY        THE CA   RIV $.
```

in which "by" is taken as correlator 0247, which is defined as '<u>Efficient Agent</u>', and

```
P 020  +-----------------------------------------2050N-+
       .                                               .
       .                                               .
P 017  .       +-------------------------------2570N-+
       .       .                                       .
       .       .                                       .
P 013  .       .       +-------------------0245E-+
       .       .       .                               .
       .       .       .                               .
P 011  .       .       .                  +-5010N-+
       .       .       .                  .            .
       .       .       .                  .            .
    JON     WAS BE  KIL VP  BY        THE CA   RIV $.
```

in which "by" is taken as correlator 0245, which is defined as '<u>Spatial Proximity</u>'.

Both parsings are correct, because the sentence is genuinely ambiguous.

APPENDIX I-d

NON-SENTENCES: Syntactically Correct Interpretations that Cannot Stand by Themselves

The syntactic interpretation of a string of words need not necessarily be the same when the string constitutes a sentence and when it constitutes only part of a sentence.

For instance the string "the rabbit we let run", presented as a sentence, has to be interpreted as an inversion of the sentence "we let the rabbit run"; if it is found as part of the sentence "we kept the squirrel but the rabbit we let run", it must be interpreted in the same way; but if it occurs as part of the sentence "they shot the rabbit we let run", it must be recognised as a relative clause.

Thus, although the interpretation as a relative clause is correct and, indeed, required under certain circumstances, we can provide for it to be excluded as a sentence interpretation when the string stands by itself.

This is implemented by a type of rule which marks certain 'complete' products as NON-SENTENCE. This comment is printed beside the product and the corresponding record prevents the product from being printed out as a graphic display.

These 'product selection' rules are triggered by the correlator number of the largest correlation in the 'complete' product, and they can involve a number of different conditions.

Some of these conditions are:

- a question mark terminating the string;
- the first correlatum of the largest correlation being a single word;
- the first correlatum of the correlation containing the first word of the input sentence;

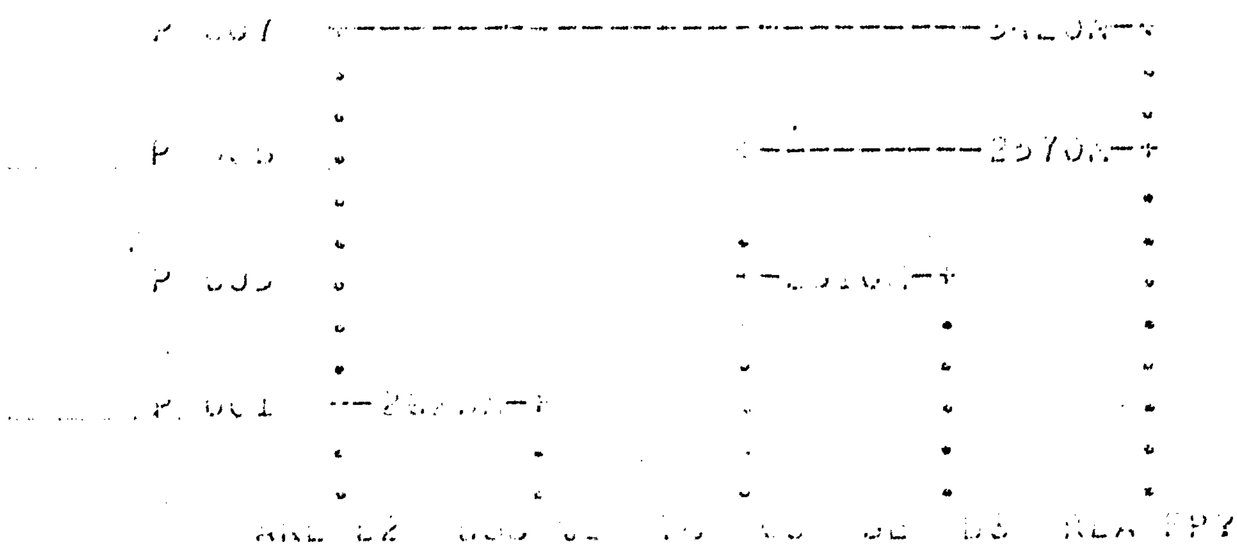
or a combination of these conditions.

The above example ("[illegible] we let run") is the simplest case and an unconditional rule can be formulated:

'A <u>complete</u> product made by one of the correlations representing a relative clause can be marked NON-SENTENCE.'

A more complex example, giving rise to two different non-sentence interpretations, is the string "are books to be read?" (see print-out on the next three pages).

Product 0007 is the correct interpretation and is displayed like this:



Note that this structure is ambiguous because correlator 3420 has not yet been univocally defined; it can mean:

a) ought books to be read?

b) will books be read? (e.g. at the meeting).

In spoken English this ambiguity is eliminated by stress. We have no way of resolving it in written text (if the sentence does not continue); but for translation, for instance into Italian, disambiguation would be indispensable because output for <u>a</u> should be "libri sono da leggere?" while output for <u>b</u> would have to be "si leggeranno libri?".

The correlational possibilities of the input string, however, are not exhausted by product 0007. The system records three more products, two of which are 'complete' but are marked as 'non-sentences' (see page I-d,4).

01 ARE	02 BOOKS	03 TO	04 BE	05 READ	? 06	07
09	10	11	12	13	14	15

P 0001	W 0002 02 [illegible]	2020·M	W 0001 01 B2		01 03	
P 0002	W 0001 01 B2	4210 N	W 0002 02 [illegible]		01 03	
S 0003	P 0002	0750 E	W 0003 01 0/		01 04	
S 0004	W 0002 02 [illegible]	0750 E	W 0003 01 0/		02 04	
P 0005	W 0003 03 00	2510 N	W 0004 01 BS		03 05	
P 0006	P 0005	2570 N	W 0005 03 FP	?	03 06	
P 0007	P 0001	3420 N	P 0006	?	01 06	COMPLETE

01 ARE	02 BOOKS	03 TO		04 BE	05 READ	? 06	07
09	10	11		12	13	14	15
P 0008	W 0002 02 %@		3850 N	P 0006		? 02 06	
P 0009	W 0001 01 B2		4210 N	P 0008		? 01 06	COMPLETE (NON SENTENCE)
P 0010	P 0005		2020 M	W 0001 01 B2		? 01 06	COMPLETE (NON SENTENCE)

Product 0009 is 'complete' but marked as 'non-sentence' and therefore does not generate a structure display. Its structure, however, can be determined by tracing its composition in the product list (see page I-d,3). The display would look like this:

```
P 0009  +------------------------------- 4210N-+
P 0008  :         +--------------------- 3650N-+
P 0006  :         :         +----------- 2570N-+
P 0005  :         :         +--2310N--+        :
        :         :         :         :        :
        :         :         :         :        :
       ARE      BOOKS       TO        BE      READ ?
```

This is the 'predicative' construction (correlator No. 4210 in the top correlation) that would be the correct interpretation in a sentence such as "all these are books to be read".

Product 0010 is also 'complete' but marked as 'non-sentence'. The display of its structure would look like this:

```
P 0010  +-2020M--------------------------------+
P 0008  :         +--------------------- 3650N-+
P 0006  :         :         +----------- 2570N-+
P 0005  :         :         +--2310N--+        :
        :         :         :         :        :
        :         :         :         :        :
       ARE      BOOKS       TO        BE      READ ?
```

This is the structure that would be correct if the input string were the subject-auxiliary phrase (correlator No. 2020 in the top correlation) in a question such as "are books to be read necessarily boring?".

Note that both the above structures contain P 0008, the noun phrase "books to be read", which is made by correlator No. 3650 (defined as '<u>Purpose</u>, expressed by infinitive').

APPENDIX II

Operative Vocabulary

Voc.Nº	Word	S	G-code	
0020	a	01	CA	indefinite article
0080	am	01	B1	auxiliary
0100	an	01	CA	indefinite article
0140	answer	01	VS	supine
		02	$-	singular count-noun
0160	answered	01	VP	past participle
		02	V8	past tense
0200	answers	01	V3	3rd person
		02	(-	plural count-noun
0220	are	01	B2	auxiliary
0380	be	01	BS	aux. supine
		02	B9	aux. subjunctive
0420	been	01	BP	aux. past participle
0440	being	01	BG	aux. pres.participle
		02	$-	singular count-noun
0500	book	01	VS	supine
		02	$-	singular count-noun
0580	books	01	V3	3rd person
		02	(-	plural count-noun
0660	bright	01	1A	adjective
0800	by	01	0/	correlator (prep.)
		02	0M	adverb
0900	can	01	M4	modal
		02	VS	supine
		03	$-	singular count-noun
0980	car	01	$-	singular count-noun
1060	Charles	01	$0	sing. non-count-noun
1280	day	01	$-	singular count-noun
1340	difficult	01	1F	framing adjective
1530	eager	01	1F	framing adjective
1580	easily	01	6M	descriptive adverb

Voc.Nº	Word	S	G-code	
1720	English	01	1A	adjective
		02	$=	plur.non-count-noun
2060	go	01	VS	supine
2100	going	01	VG	present participle
2140	gone	01	VP	past participle
		02	1A	adjective
2240	had	01	H8	auxiliary, past t.
		02	HP	aux.past participle
2340	has	01	H3	aux. 3rd person
2360	have	01	HS	aux. supine
2400	he	01	$1	personal pronoun
2500	her	01	A3	accusative pronoun
		02	CP	possessive adjective
2620	him	01	A3	accusative pronoun
2660	his	01	CP	possessive adjective
		02	+P	possessive pronoun
2780	I	01	$1	personal pronoun
2900	in	01	0/	correlator (prep.)
		02	OM	adverb
2920	is	01	B3	aux. 3rd person
2940	it	01	$3	personal pronoun
			$9	impersonal pronoun
2990	Jones	01	$0	sing.non-count-noun
3070	killed	01	VP	past participle
3080	kind	01	1A	adjective
		02	$-	singular count-noun
3260	knows	01	F3	framing v.,3rd person
3320	larger	01	2A	comparative adjective
3430	learn	01	FS	framing v., supine
3500	letters	01	(-	plural count-noun
3700	likely	01	1F	framing adjective
3780	little	01	CQ	determiner
		02	$Q	pronoun

Voc.N°	Word	S	G-code	
	little (continued)	03	1Q	adjective
		04	6Q	adverb
4010	mad	01	1A	adjective
4220	me	01	A1	accusative pronoun
4460	my	01	CP	possessive adjective
4780	on	01	0/	correlator (prep.)
		02	OM	adverb
4800	one	01	CN	numeral
		02	$N	pronoun
		03	$I	impersonal pronoun
		04	$-	singular count-noun
5320	read	01	FS	framing v., supine
		02	$-	singular count-noun
		03	FP	framing past participle
		04	F8	framing v., past tense
5340	reading	01	FG	framing pres. participle
		02	$-	singular count-noun
5380	reads	01	F3	framing v., 3rd person
5390	red	01	1A	adjective
		02	$=	sing.non-count-noun
5400	river	01	$-	singular count-noun
5420	sat	01	V8	past tense
5540	she	01	$1	personal pronoun
5750	speak	01	VS	supine
5790	Sunday	01	$-	singular count-noun
5980	the	01	CA	definite article
6100	they	01	(3	personal pronoun
6180	three	01	CN	numeral
		02	$-	singular count-noun
6320	to	01	0/	correlator (prep.)
			00	particle
6400	train	01	FS	framing v., supine
		02	$-	singular count-noun
		03	VS	supine (intransitive)

Voc.N°	Word	S	G-code	
6420	trained	01	FP	framing past participle
		02	F8	framing v., past tense
6460	trains	01	F3	framing v., 3rd person
		02	(-	plural count-noun
		03	V3	3rd person (intransitive)
6800	wanted	01	FP	framing past participle
		02	F8	framing v., past tense
6860	was	01	B6	aux., past tense
6880	we	01	(1	personal pronoun
6900	went	01	V8	past tense
6920	were	01	B8	aux., past tense
		02	B5	aux., subjunctive
7520	you	01	+2	personal pronoun

Note: This vocabulary contains the words with which we are at present experimenting (with a view to further correlation restraints and to the introduction of punctuation marks); it does in no way reflect the capacity of the program. It is being kept at a minimum in order to avoid repetitious card punching when corrections and alterations are being tested.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Georgia Institute for Research 711, C & S Bank Bldg. ATHENS, Ga., 30601	2a. REPORT SECURITY CLASSIFICATION 2b. GROUP

3. REPORT TITLE

THE MULTISTORE SYSTEM MP-2

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Scientific Progress Report

5. AUTHOR(S) *(First name, middle initial, last name)*
Ernst von Glasersfeld
Pier Paolo Pisani

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
15 November, 1968	67	9

8a. CONTRACT OR GRANT NO. AFOSR 1319-67 b. PROJECT NO. 9769-05 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) MP-2 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Air Force Office of Scient. Research Directorate of Information Sciences Arlington, Virginia, 22209

13. ABSTRACT

The report describes procedure and machine program of the second version of the Multistore Sentence Analysis System implemented on an IBM 360/65. Using a correlational grammar (described in previous reports) the systems parses English sentences and displays the parsings as hierarchical syntactic structures comparable to tree-diagrams. Since correlational syntax comprises much that is usually considered semantic information, the system demonstrates ways and means of resolving certain types of ambiguity that are frequent obstacles to univocal sentence analysis. - Particular emphasis is given to the 'significant address' method of programmimg, developed to speed up the procedure (processing times, at present, are 0.5-1.5 sec. for sentences up to 16 words). By structuring an area of the central core in such a way that the individual location of bytes becomes significant, the shifting of information is avoided; the use of binary masks further simplifies the many operations of comparison required by the procedure.

Samples of print-out illustrate some salient features of the system.

DD FORM 1 NOV 65 1473

unclassified
Security Classification

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Parsing						
Natural-Language Analysis						
Syntactic Analysis						
Semantic Analysis						
Computational Linguistics						
Machine Translation						
Programming (machine language)						
Matching Procedures						
Pattern Recognition						

unclassified

Security Classification