The Multistore Parser for Hierarchical Syntactic Structures

ERNST VON GLASERSFELD
AND
PIER PAOLO PISANI
The University of Georgia and
The Georgia Institute for Research, Athens, Georgia

A syntactic parser is described for hierarchical concatenation patterns that are presented to the analyzer in the form of linear strings. Particular emphasis is given to the system of "significant addresses" by means of which processing times for large-scale matching procedures can be substantially reduced. The description makes frequent use of examples taken from the fully operational implementation of the parser in an experimental English sentence analyzer.

By structuring an area of the computer's central core storage in such a way that the individual locations of bytes and bits come to represent the data involved in the matching procedure, the shifting of information is reduced to a minimum, and the searching of lists is eliminated altogether. The matches are traced by means of binary masks and the state of single bits determines the operational flow of the procedure. The method could be implemented with any interpretive grammar, provided it can be expressed by the functional classification of the items composing the input hierarchical structures.

KEY WORDS AND PHRASES: parsing, syntactic analysis, natural-language analysis, linguistic data processing, computational linguistics, correlational grammar, structure recognition, pattern recognition, matching procedures, tree-structure interpretation, machine translation, automatic abstracting
CR CATEGORIES. 3.42, 3.49, 3.63, 3.70, 3.71

Introduction

The Multistore Parser was developed in the course of a research project concerned with automatic analysis of natural language sentences. It is an attempt to keep within reasonable (i.e. practically useful) bounds the processing times involved in a matching procedure that has to cope with an extremely large amount of data, such as is encountered in structures made up of relatively simple but combinatorially highly versatile items whose hierarchical concatenation is governed by an intricate but specifiable grammar.

The parser was first implemented (MP-1) on a GE 425 [1] and is now fully operational in a second version (MP-2) on an IBM 360/65 [2]. Its speed and efficiency have turned out to be very satisfactory, and the fact that it gives the user a great deal of freedom regarding the alteration and refinement of the incorporated grammar makes it seem plausible that the system could be successfully applied to other problems requiring a large-scale matching procedure.

By *parsing* we intend the recognition and explication of concatenation patterns that conform to a system of rules (GRAMMAR) and are presented to the analyzer as a string (SENTENCE) of consecutive units (WORDS) which correspond to individuals of a pre-established set (VOCABULARY).

The *grammar* can be divided into four parts which entail operational differences. The first is a collection of connective or syntactic functions (CORRELATORS) which constitute the linking element in any CORRELATION.[1] The grammar provides for each correlator an individual CORRELATOR NUMBER (CN). Since it may be possible (as it is, for instance, in natural language sentences) to distinguish different types of correlator, whose connective function applies in characteristic ways and therefore has a bearing on the MODE of correlation, a code indicating the CORRELATOR TYPE (CT) is added to the correlator number CN.

(In English sentences, for instance, one finds correlators that are indicated by a specific word, i.e. EXPLICIT correlators, and others that are indicated only by the juxtaposition of the correlated pieces, i.e. IMPLICIT correlators; this difference is exemplified also in the mathematician's practice of sometimes explicitly indicating 'multiplication' by "a × b" and at other times indicating it implicitly by writing "ab". Another distinction of CT reflects the sequence of the linked pieces; thus, in English, the correlator that links "John has" is basically the same as the one that links the inversion "has John", but since the sequence is relevant to the parsing and affects the mode of correlation, it is convenient to split the connective into two correlators bearing the same CN but different CT.)

The correlator list of the grammar must also provide a pragmatic description of the relation embodied by each correlator; this description can be an ad hoc characterization, an illustrative paraphrase, a suitable transform à la Chomsky [5] or Fillmore [6], or a symbolic expression devised along the lines of function symbols in logical calculus [7]. The kind of characterization chosen is irrelevant

The research reported in this paper was sponsored by the Air Force Office of Scientific Research under Grant AFOSR 1319-67 (Information Sciences Directorate).

[1] The bases of "correlational grammar" were laid by Silvio Ceccato during his investigation of human thought processes [3]; preliminary applications to natural language, i.e. Italian, Russian, and English, were drafted by Zonta, Perschke, Barton Burns and v. Glasersfeld at the Center for Cybernetics, Milan University, Italy [4].

to the recognition procedure of the parser; it serves only in the explication of the discovered patterns. What type of explication will be considered the most useful depends, of course, on the kind of phenomenon we are analyzing.

The second part of the grammar is the classification of *vocabulary* items in terms of their individual capacity to function as components of correlations. Such a classification is achieved by assigning to each word a string of indices, each of which indicates the number CN of a particular correlator by means of which the word can be correlated to something else, as well as the role it can play in that correlation, i.e. its CORRELATIONAL FUNCTION (CF).

The complex code CN + CT + CF, which specifies *one* correlational possibility of a given word, is called CORRELATION INDEX or Ic. For reasons of procedural convenience the string of Ic's, which specifies *all* the correlational possibilities of a vocabulary word, is divided into substrings according to the CT and CF of the individual Ic's.

Since the words of a vocabulary may not always be univocal (i.e. they may be homographs and have more than one sense, as for instance nearly all the words in an English vocabulary), their different senses must be distinguished and recorded in the vocabulary; this is essential, because a parsing procedure can resolve the initial ambiguity of a homograph whenever only one of its senses fits the correlational structure of the given sentence (the frequency and importance of this kind of automatic disambiguation is, of course, proportional to the refinement of distinctions in the list of correlators). Clearly separable senses of a word are, therefore, discriminated by a code number S in the vocabulary.[2] Separate S-values are treated as different words WS in the procedure, except for the fact that they occupy one and the same place in the word sequence of a given sentence.

The third part of the grammar concerns RECLASSIFICATION. Since the structures to be recognized by the parser are hierarchical, or tree-like, they must contain correlations whose components are themselves the results, or PRODUCTS, of correlation. Hence products must, like single words, be assigned Ic's before they can play the role of first or second piece in a wider correlation. But whereas the Ic-strings of WS's in the vocabulary can be assigned a priori, the Ic-strings of products cannot, because they depend not only on the correlator responsible for the product but to some extent also on the particular pieces that happen to be linked in the given instance of that correlation.[3] In other words, each correlator determines the general lines of Ic assignation to its products, but the individual Ic's to be assigned are often determined by the product's specific components.

For this reason the reclassification procedure takes place in two steps: each correlator is provided with a LIST (RL) which indicates a set of RULES (RR) which specify a string of Ic's as well as the conditions under which these Ic's are to be assigned to the product.

The fourth part of the grammar is inherently nongeneralizable because it consists of the collection of ad hoc rules required by the specific application of the system, i.e. rules concerning aspects of the input that have a bearing on the parsing but for one reason or another elude the basic Ic classification. In MP-2, the English sentence parser, for instance, special routines had to be added for the recognition of idiomatic phrases; for the elimination of parsings of a given sentence that would be correct if the sentence were longer—and, therefore, in expectation have been foreseen by the system—but are unacceptable for the sentence as it is; and for some other peculiarities of natural language input that are (or still seem) unsystemic.

The number of operative correlators, in MP-2, is approximately 350 and the number of Ic's assigned to one WS in the vocabulary ranges from 3 to 136. Taking into account that reclassification assigns, on an average, 45 Ic's to each product made during the analysis of a sentence, it becomes clear that the amount of matching operations to be carried out before a coherent correlational structure is reached that comprises all the words of an input sentence of say, 15 words, is very large indeed. It was, therefore, essential to make the single matching operation as fast as possible by avoiding any shifting of the data involved. This was achieved by giving each correlator a fixed sequence of addresses (fixed, that is, relative to one point of origin) in an area MS of the computer's central core, so that all matching operations concerning Ic's of one specific correlator could be done by scanning that same small section of addresses. Each sequence of addresses that represents one correlator was then structured in such a way that each of its bytes came to represent one specific piece of the input sentence; thus the need to shift these two types of information was eliminated throughout the matching procedure, because, at any given moment, the address of the location where we are operating implies both (i.e. both the correlator *and* the specific piece with which we happen to be concerned). As we shall see later, the same MS-area, once structured in this two-dimensional way, serves as well to avoid information shifting during the reclassification routines.

The Multistore Area MS is best visualized as a rectangular area structured in vertical COLUMNS (each representing one correlator) and horizontal LINES (each representing one piece with Ic classification).

[2] Separability of senses is determined by at least one incompatible Ic-assignation. In an English vocabulary, for instance, the word "can" must be split into at least three S's:

can, 1	modal auxiliary,
can, 2	verb (=to pack in cans),
can, 3	noun (=container).

Some of the incompatibilities in this case are: "can, 1" must be assigned an Ic *a* for the construction "John can sing", which cannot be formed with either "can, 2" or "can, 3"; whereas "can, 2" must be assigned an Ic *b* for the construction "we can strawberries", which cannot be formed with either "can, 1" or "can, 3"; etc.

[3] For a detailed discussion of 'reclassification' in English sentence analysis cf. [1, ILRS T-11, pp. 27–36; ILRS T-14, pp. 1–9].

Pre-established Data

The VOCABULARY, containing all the WORDS the system can handle, is stored in a disk file. Since words can be split into different senses, word senses (WS) are the basic unit of the vocabulary.

Each WS-entry consists of:

—a vocabulary number, i.e. the number of the word to which the WS belongs. In MP-2 vocabulary numbers reflect the alphabetic order of the words contained in the vocabulary;

—an S-code, which distinguishes the different senses WS belonging to one and the same word;

—a GT-code, indicating general grammatical and semantic characteristics;

—an alphanumeric representation of the word to which the WS belongs; (this plays no part in the parsing procedure but serves to make output more easily readable;)

—the Ic-string assigned to the WS, divided into substrings according to the Ic's combination of CT and CF. (Note that every Ic is a complex code consisting of CN + CT + CF.)

The CORRELATOR TYPES operative in MP-2 are six, three IMPLICIT and three EXPLICIT. Each has its individual MODE of correlation and requires an obligatory sequence of CF's in the sequence of pieces to be correlated.

The three IMPLICIT correlator types and the CF-sequences required by them are:

TYPE N: implicit, no inversion (the pieces occur in their 'normal' order in the input sequence);

e. g. word order:	"John	has"
CT-CF's:	N1	N2
product:	P(N)	

TYPE M: implicit, inversion;

e. g. word order:	"has	John"
CT-CF's:	M2	M1
product:	P(M)	

TYPE V: implicit, no inversion, obligatory comma;

e. g. word order:	"Ithaca,	a Greek island, ··· "	
CT-CF's:	V1	,	V2
product:	P(V)		

The pieces bearing N1, M2, or V1, are LEFT-HAND pieces (LH), while pieces bearing N2, M1, or V2, are RIGHT-HAND pieces (RH).

In order to keep the correlation procedure as homogeneous as possible, explicit correlations are formed in two steps; the first combines two of the three pieces to form a SEMIPRODUCT (SP); this SP is automatically assigned an "intermediary" function CF4, by means of which, in the second step, it is combined with the third piece.

The three EXPLICIT correlator types and the CF-sequences required by them are:

TYPE E: explicit, no inversion;

e. g.	word order:	"strangers	in	Paris"
	CT-CF's:	E1	E3	E2
	semiproduct:	SP(E4)		
	product:	P(E)		

TYPE F:

word order:	"in	water	they float"
CT-CF's:	F3	F2	F1
semiproduct:	SP(F4)		
product:	P(F)		

TYPE H:

word order:	"cats	she plays	with (but dogs she won't touch)"
CT-CF's:	H2	H1 H3	
semiproduct:		SP(H4)	
product:	P(H)		

In the formation of an SP, therefore, the procedure takes pieces bearing E1, F3, or H1, as LEFT-HAND pieces, and pieces bearing E3, F2, or H3, as RIGHT-HAND pieces; in the second step, combining SP's with the remaining item of the correlation, E4, F4, and H2 represent LEFT-HAND pieces and E2, F1, and H4 represent RIGHT-HAND pieces.

The operative CT-CF code combinations, which characterize individual Ic's and determine the substring division in MP-2, are:

N1, M1, V1, E1, F1, H1,
N2, M2, V2, E2, F2, H2,
E3, F3, H3.

Since no piece can be assigned Ic's of all 15 substrings, the length of the total Ic-string is left variable. (The greatest number of substrings shown by one WS in the present vocabulary is 9, and the average number of Ic's in one substring is 6.)

The pre-established data concerning the RECLASSIFICATION of products are expressed in lists RL and rules RR, which are recorded in MS-columns and MS-lines respectively.

Each RL contains:

—the CN-CT of the correlator the reclassification of whose products it concerns; this CN-CT is the address of

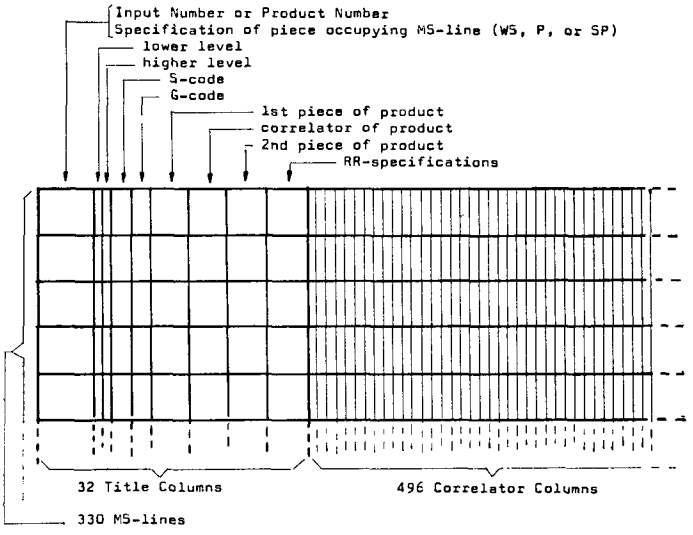


FIG. 1

the MS-column that is dedicated to that same correlator;

—the set of rules RR which govern the reclassification of that correlator's products; each member of that set is indicated by a "rule-marker" (bit 5, Figure 2) in the byte that constitutes the intersection of the correlator's MS-column with the MS-line dedicated to the relevant RR.

Each RR contains:

—an implicit label or number, which is, in fact, the address of that RR in the 'title columns' (see Figure 1) of the MS-area, and thus indicates the MS-line dedicated to that RR;

—the substring of Ic's assignable by that RR; each of these Ic's is recorded by means of an 'assignation marker' (bits 6 and 7, Figure 2) in the byte constituting the intersection of the rule's MS-line with the MS-column of the correlator to which the individual Ic's CN-CT code refers;

—an indication of the CF of the substring's Ic's;

—the specific conditions under which the RR assigns the Ic's indicated in its substring.

Multistore Area

Represented as a rectangular area, MS has as many vertical columns (one byte wide) as there are operative correlators. As in a matrix, each byte of a column belongs also to a horizontal line, which is one byte high. Each MS-line, besides, has a number of "title bytes" (see Figure 1) at its beginning, on the left, which serve to specify what the main part of the line contains. The specifications may be:

—input number, S-code, GT-code, and level-indication (cf. below), if the occupant is a single WS;

—product number, addresses of the components, GT-code, and level-indication, if the occupant is a product or semiproduct;

—and, as separate, pre-established information, the specification of the RR occupying the line.

The remaining part of the line represents the Ic-strings of the occupant by means of Ic-markers in those bytes that also belong to the columns of the correlators to which the individual Ic's refer. In the case of WS's and products, these markers are placed in the first four bits (bits 0, 1, 2, 3) of the relevant byte and are instrumental in the correlation procedure; in the case of RR's the markers are placed in bits 6 and 7 of the relevant byte and indicate that the Ic represented by the column to which the byte belongs is to be considered for reclassification by the RR occupying that line.

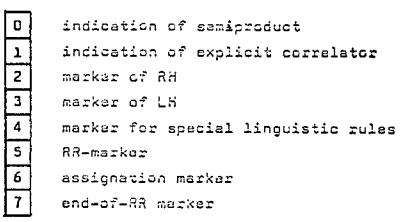


FIG. 2

Each MS-column, on the other hand, is dedicated to one specific correlator, and the column's address represents that correlator's CN-CT. As pre-established information each column also contains the RL of the correlator, i.e. the bytes that constitute intersections with the lines dedicated to RR's belonging to that specific list contain an RR-marker (bit 5).

The Ic's function CF determines the mode of operating within the column specified by the Ic's CN-CT. This operational characteristic of an Ic is represented by the specific configuration of bits constituting the Ic's marker within the relevant column.

Direction of Parsing and Correlation Levels

Both Correlational Grammar and the Multistore System were devised expressly for the parsing of input that has the form of a linear string of 'words'. The significance of sequence may vary a great deal in different applications, but it is difficult to conceive of complex item structures that are conventionally or habitually represented as linear strings in which the sequence of items has no significance at all. In natural language it certainly has,[4] and in the MS-parser, therefore, the sequence of words given by the input sentence plays an important part. MP-2, like the human reader of English, works from left to right. This, of course, is not mandatory, but since the most difficult part of sentence analysis is the isolation of the criteria by means of which the human reader determines interpretations and resolves ambiguities, it was felt that maintaining the direction unchanged would help both the discovery and the implementation of these criteria—at least in the first attempt. (Whether or not this is, in fact, the most economical way of proceeding, could be established only by an actual comparison of left-to-right, right-to-left, and mixed procedures.)

In the MS-area the input sequence is reflected by the succession of lines from the top down. The first WS of the first word occupies the topmost MS-line, the second WS the second line, and so on. All the WS's of the first word, together, constitute the first LEVEL; the WS's of the second word belong to the second level, and so do any products composed of words one and two. The level of a WS is determined by its place in the input sentence; the level of a product is determined by the last WS it contains as component. For reasons of operational homogeneity, the LEVEL-INDICATION assigned to every piece in the title section of MS consists of two data: its lower level and its upper level. For a WS corresponding, for instance, to the third word of the sentence, the level-indication is 3,4; for a product composed of words two, three, and four, the level-indication is 2,5.

[4] The significance of sequence varies considerably in natural languages; it is great in Chinese and English, minimal in Latin. Note, also, that in the case of a conjunction such as "and" (where, in logic, the terms are absolutely interchangeable) the sequence may be subject to semantic restrictions in language (e.g. "Mary married John and had five children").

Procedure

The WS's corresponding to the words of the input sentence[5] are read into the MS-area so that each WS occupies one line. The title section of the line receives the specifications of the WS, and for each Ic contained in the WS's substrings a marker is inserted into the corresponding byte of the line, i.e. the byte that forms the intersection of the line with the column dedicated to the particular Ic's CN-CT. This insertion is *immediate*, since the Ic itself is the address of the relevant column. The configuration of the marker indicates the Ic's CF.

The CT-CF of the Ic in hand determines the correlational mode. If it represents an LH, its marker remains passive, i.e. it triggers no correlation routine. If it represents an RH, insertion of the marker triggers an upward search (through the previous levels of the column) for a complementary marker, i.e. an LH-marker. If such a marker is found, the level of the piece it represents is checked (level-indication contained in the title section of the marker's MS-line), and if the highest level covered by it is contiguous with the lowest level covered by the RH from which the search started, a product is recorded on the next free MS-line.

Composition of the P's record in the title section of the new line is immediate, because all the data required for it are implicit in the operational path that led to its production. The LH component of the product correlation is the piece represented by the found marker, and the address of that marker's MS-line is now inserted into the relevant title bytes of the new P's line; the CN-CT of the correlator responsible for the P is the address of the column which was searched, and this address is now inserted into the relevant title bytes of the P's line; the RH component is the piece represented by the marker that triggered the search, and the address of its MS-line constitutes the third element of the new P's record. Finally, the new P's level-indication is compiled by recording the lowest level of the P's LH and the highest level of its RH.

When the new P's specification in the title section of its line is complete (its Ic-strings are supplied later) insertion of markers for the Ic's of the piece that caused the P's production is resumed and any further products resulting from these insertions are recorded in the same manner on subsequent free MS-lines.

Reclassification of the recorded products takes place after the last Ic contained in the substrings of the piece that caused production has inserted its marker. The first new product record (i.e. the uppermost one that, as yet, has no Ic-string) is now examined in that part of its title section that indicates its correlator. The code found there is the address of the MS-column dedicated to that correlator. That column is then scanned, from the top down, for RR-markers (bit 5); when a marker is found, it indicates by its position that the MS-line to which it belongs contains an RR relevant to the reclassification of the P in hand. The title section of the line specifies the conditions of the rule, and the CT-CF of the Ic's it is to assign; if the conditions are satisfied, the line is scanned for assignation markers (bit 6) and the RR's end-sign (bit 7). Every assignation marker indicates that the CN represented by the column it is found in is the CN of an Ic to be assigned to the product. (The CT of that Ic is also implicit in the column, and the CF was given in the title of the RR.)

Since the column represents that same CN-CT in all its intersections with MS-lines, assignation of the Ic to the product is immediate: it takes place on the line representing the product and it is in every way similar to the insertion of Ic-markers originating from the substrings of WS's. If the insertion of these markers causes further production, the new P's are recorded in the same manner as those springing from WS-marker insertion; after the last of these new P's has been recorded in the title section of a free line, reclassification of the P in hand is resumed.

Output

When production has ceased, that is to say, when the last WS corresponding to the last word of the input sentence has inserted all its markers and all P's resulting from these have, in turn, been reclassified and gone through their own productive cycles, the output phase is reached.

The primary output consists of graphic representations of all products that contain all words of the sentence. The representation is equivalent to a tree-structure displaying the WS's at the terminals and the operative correlators (i.e. their CN-CT code) at the nodes (see Appendix).

This graphic display is compiled by first scanning the title columns that contain the level-indication of P's (complete P's for a sentence of n words will show the indication 1, n + 1); the composition of a complete P is then traced by means of the addresses (in the product-specification columns) which refer to its components, and from these to the components' components, and so on, until the relevant WS's are reached.

Since MP-2 was written to serve as basis for the further development of correlational grammar and semantic disambiguation of English sentence parsings, two secondary types of output were added as an aid to linguistic research: a list specifying all products recorded during the analysis of the given sentence, and a complete listing of every product's reclassification.

Technical Data

The size of the MS-area implemented in MP-2 is 528 columns by 330 lines, making a total of 174,240 8-bit bytes. The entire system, including conversion table (linguist's Ic-code into MS-addresses and vice versa), accessory procedures for idiomatic phrases and special linguistic rules, buffer area for word input, and complete program, occupies approximately 190k of the IBM 360/65's 500k central core.

[5] The step from the words of the sentence to the corresponding WS's in the system's vocabulary can be implemented by any kind of 'look-up' procedure and its organization is irrelevant to the system we are describing here.

At present the system operates with some 350 correlators and 300 reclassification rules; given recent refinement of correlational grammar, we expect MP-3, the next implementation of the parser, to operate with approximately 450 correlators and 400 RR's, yielding considerably greater specificity of the parsings with regard to certain syntactic areas (prepositional relations, adjectival functions, and transitivity).

The system accepts sentences of up to 16 words and the average processing time required for the complete parsing is 0.2 sec per word.

The program for MP-2, with the exception of accessory instructions concerning card-reader, disk file, and printer, was written in basic machine language.

Conclusion

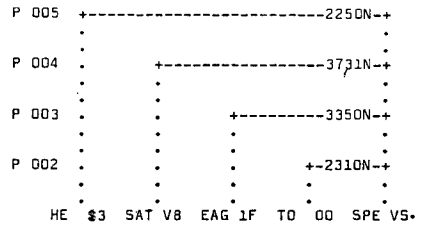
Although the Multistore Parser was developed as a result of the application of correlational grammar to sentence analysis, our exposition should have made it clear that it could be used to serve almost any kind of grammar, provided that the requirements and rules of the grammar can in some way be expressed by a classification of the single items whose concatenation the grammar governs. The possibility of providing for different correlator types and different operational modes of correlation makes the system capable of implementing widely diverse grammars (ranging from those based on traditional syntax to experimental ones incorporating semantic association). Moreover, the fact that in MP-2 "explicit" correlations are arrived at by a procedural succession of two binary combinations does not mean that the system could not handle ternary relations; an early version did this quite successfully (by splitting the MS-columns of the relevant correlators into three subcolumns), but the procedure was given up, because, given the 8-bit byte of the machine available the two-step procedure was more economical as to space.

With regard to the use of the machine's storage capacity, there is one essential point we should like to stress. The general trend in linguistic computer applications has for some time been determined by the idea that, given the vast amount of data involved, it was inevitable that the mass of operative data be kept on accessory storage devices (tapes, disks, drums). The Multistore Parser, on the other hand, sprang from the assumption that the development of computers would, in any case, lead to gigantic central cores (either as such, or with the help of Additional Core Storage) and that it was by adequately structuring the use of these, rather than of larger but slower accessories, that language analysis (or, indeed, any interpretive procedure for very large data bases) could attain real-time usefulness.

Appendix

The samples of output, i.e. graphic representations of the correlational structures of input sentences, given in Examples I–IV are taken from routine jobs run in the spring of 1969. The samples of print-out were copied on a typewriter because actual print-out proved too light for successful reproduction. For an explanation of the correlators involved in the examples see the list at the end of this Appendix.

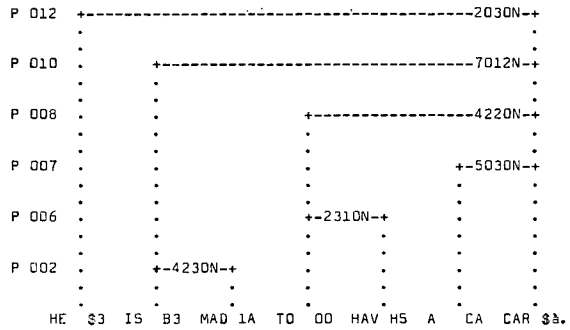
Example I. Input sentence: "He sat eager to speak"



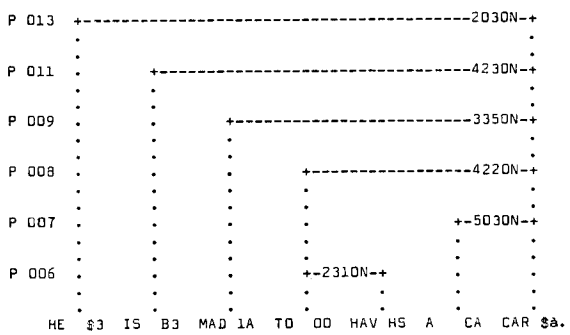
Example II. Input sentence: "He is mad to have a car"

Note that "mad // to have ···" (P:009) is linked by the same correlator (3350 N) as "eager // to speak", P:003 in Example I.

Interpretation (b) springs from the second sense of the adjective "mad" (i.e. mad = desirous) which is linked to "to have a car" (P:008) by a different correlator, namely 7012 N; this second structure can be paraphrased: 'it is mad *of him* to have a car'. Since the sentence is genuinely ambiguous, both interpretations are correct and necessary.



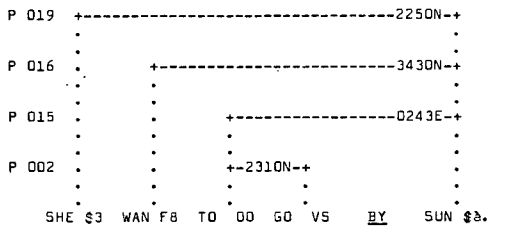
Interpretation (a)



Interpretation (b)

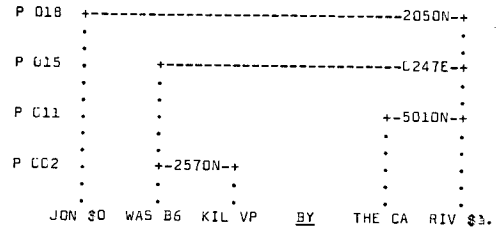
Example III. Input sentence: "She wanted to go by Sunday"

Although the preposition "by" has been split into ten subrelations, each of which has a correlator to itself (see note under 0247 E in the correlator listing), this sentence correctly yields only one interpretation; the correlation made by "by" (P:015) shows correlator 0243 E, which specifies the point in time after which the assertion preceding it is to be considered an accomplished fact.

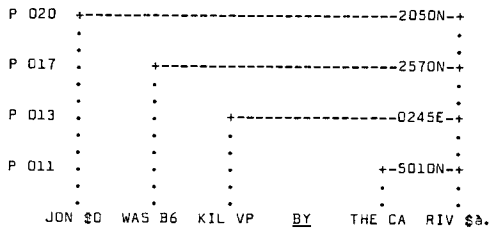


Example IV. Input sentence: "Jones was killed by the river"

In this sentence two interpretations are correct and the system produces interpretation (a) in which "by" is taken as correlator 0247, which is defined as *'Efficient Agent'*, and interpretation (b) in which "by" is taken as correlator 0245 E, which is defined as *'Spatial Proximity'*.



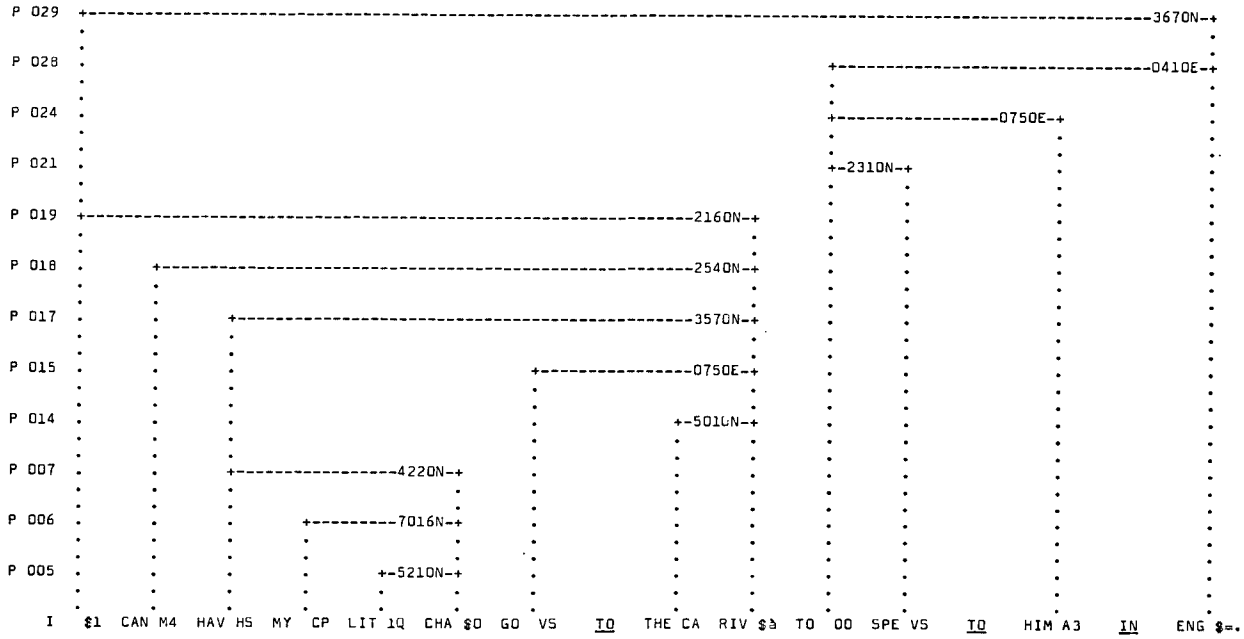
Interpretation (a)



Interpretation (b)

Example V. Input sentence: "I can have my little Charles go to the river to speak to him in English"

To illustrate the full capacity of the present system, we give one example of a 16-word sentence. Except for the specification of the relations expressed by the prepositions "to" and "in" (not yet incorporated in the system) the univocal interpretation of the sentence can be considered definitive.



Special Terms, Signs, and Abbreviations

'Substantive'—the term is used to indicate any word or word combination that can play the part of *subject* to a verb; "the chair", "chairs", "her little child", "going to Rome", "wine", "virtue", etc., are 'substantives'; "chair", "child", etc., are not.

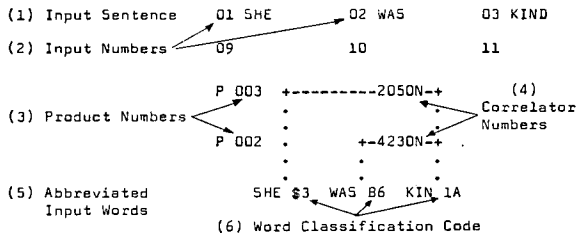
'Framing'—indicates generically what traditionally has been called 'clause-transitivity' in the case of verbs; in correlational grammar there are 'framing' adjectives and nouns as well as verbs (e.g. "The *decision* to talk", "*eager* to talk").

LH—left-hand piece of a correlation.

RH—right-hand piece of a correlation.

//—place of the correlator in a string of words.

Explanation of the Graphic Structure Display



(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 the individual words are identifiable only by their input numbers.

(3) The product numbers reflect the sequence of pro-

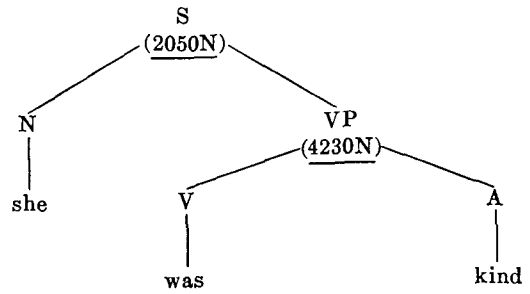
duction in the course of the analysis; the first product number, in the left top corner of the graphic display, is the number of the displayed product.

(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 stands at the beginning.

(5) At the terminals of the structure the three first letters of the relevant word are printed.

(6) The word code represents a summary of grammatical and semantic data and serves mainly as an aid in examining the print-out: it tells the linguist which sense of the given input word is operative in the particular correlation he is looking at.

Note. The displays are, in fact, tree-structures, but the usual, slanting branches are represented by rectangular connections, which are less troublesome to produce on an on-line printer. What distinguishes a correlational tree-structure from those of traditional grammar is the exact labeling of the nodes. The correlator numbers, thanks to the a priori explication of correlators, indicate with considerable precision the particular semantico-syntactic relation that links the two items at the nearest lower branch ends. The above correlational structure diagram can easily be transformed into the conventional tree-structure below.



The correlator number 2050 N, however, immediately specifies that N can be nothing but a third person singular pronoun or a singular substantive, while VP must be formed with the word "was". The correlator number 4230 N specifies that V can be nothing but a form of a "predicative" verb (e.g. to be, to become, to seem) and A nothing but a predicative adjective. Given the extreme simplicity of this sample sentence, the correlational parsing adds little to the parsing by conventional grammars; correlational specificity becomes more evident in more complex sentences.

Correlators Involved in the Examples

Explicit Correlators

0243 E generic type: "BY", temporal limitation
Explication: the right-hand item specifies the point in time, after which what is asserted by the left-hand item is to be considered an accomplished fact (e.g. "I shall be back *by* Christmas", "He had lost his influence *by* 1821").
LH: perfective or future verb phrase, or predicative adjective or past participle;
RH: specific temporal term.

0245 E generic type: "BY", spatial proximity
Explication: the right-hand item serves as spatial point of reference ("landmark"), to which the left-hand item is considered proximate (e.g. "He sat *by* the window", "The house *by* the river").
LH: nonlocomotive verb, or noun;
RH: limited two- or three-dimensional item
Note: very extended two-dimensional items (e.g. "river", "sea", etc.) are exceptions since they can form the same relation also with locomotive verbs, e.g. "He walked by the river"; with items of ordinary extension, this would give rise to the relation of 'proximate passage', as for instance in "He walked by the church".

0247 E generic type: "BY", efficient agent
Explication: the right-hand item specifies the actor of the activity indicated by the left-hand item (e.g. "The answer given *by* Charles").
LH: passive form of verb (form of 'to be' plus past participle);
RH: any substantive.
Note: the system at present distinguishes seven other relations expressed by "by" (which do not occur in the examples given in this appendix); they are: Specification of Ambient Circumstance (e.g. "They played by moonlight"), Authorship (e.g. "A book by Hemingway"), Specification of Itinerary (e.g. "He arrived by the fields"), Means of Transport (e.g. "We travelled by car"), Method: Activity (e.g. "He learned it by watching professionals"), Method: Instrument (e.g. "to translate by computer"), Difference expressed as measurement (e.g. "he was taller by two inches").

0410 E generic type: "IN"
Note: no subdivisions of the relational function of "in" have as yet been inserted into the system.
LH: empirical selection;
RH: any substantive and certain specific nouns.

0750 E generic type: "TO"
Note: no subdivisions of the relational function of "to" have as yet been inserted into the system.
LH: empirical selection;
RH: any substantive and certain specific nouns.

Implicit Correlators

2030 N generic type: subject//verb
LH: "he", "she", "it", and any singular substantive;
RH: "is".[6]

2050 N generic type: subject//verb
LH: "I", "he", "she", "it", and any singular sub-

[6] In the specifications of LH and RH, items between double quotes refer to *individual* words.

stantive;
RH: "was".

2160 N generic type: subject//verb
LH: any personal pronoun, any substantive;
RH: "can", "may".

2250 N generic type: subject//verb
Explication: actor // activity carried out;
LH: any personal pronoun, any substantive;
RH: past tense of any verb.

2310 N generic type: auxiliary combination (infinitive)
LH: "to";
RH: supine form of verb or auxiliary.

2540 N generic type: auxiliary combination (modal)
LH: "shall", "should", "will", "would", "can", "could", "must", "may", "might", "do", "does", "did";
RH: "have".

2570 N generic type: auxiliary combination (passive)
Explication: grammar subject is object of activity.
LH: any form of 'to be';
RH: past participle.

3350 N generic type: clause-transitive relation
Explication: the subject is the actor of the infinitive activity; the activity is merely envisaged and the adjective specifies the subject's attitude towards it (e.g. "John is eager to please").
LH: framing adjective of type B;
RH: infinitive.

3430 N generic type: clause-transitive relation
Explication: the subject is the actor of the infinitive activity; the left-hand verb specifies the subject's relation to the execution of the infinitive activity (e.g. "They started to run", "He loves to talk").
LH: framing verb of type 343;
RH: infinitive.

3570 N generic type: clause-transitive relation
Explication: the object is the actor of the supine activity; the subject perceives, intends, or causes the object's acting (e.g. "We saw John leave").
LH: framing verb of type 357 plus object;
RH: supine form of any verb.

3670 N generic type: verb phrase specified by infinitive
Explication: the infinitive specifies the *purpose* of the subject's activity (e.g. "He works to live").
LH: subject + verb (excluding framing or predicative verbs);
RH: infinitive.

3731 N generic type: adjectival modification
Explication: the adjective specifies the subject's state during the activity (e.g. "He lay silent"; paraphrase: he lay *and was* silent).
LH: subject + verb;
RH: adjective.

4220 N generic type: verb//object
Explication: the object appertains to the subject.
LH: any form of 'to have';
RH: any substantive.

4230 N generic type: adjectival predication
Explication: the adjective is predicated of the subject of the verb.
LH: any form of a predicative verb;
RH: predicative adjective.

5010 N generic type: determiner//noun
Note: the results of a preliminary investigation of this area indicate that the diverse functions of the articles in English could be represented by seven or eight individual correlators; the subdivisions, however, have not yet been introduced into the operational grammar.
LH: "the";
RH: nouns and certain types of noun phrase.

5030 N generic type: determiner//noun
(see note under 5010 N)
LH: "a" or "an";
RH: singular count-noun.

5210 N generic type: adjectival modification
Explication: the adjective specifies a property attributed to the item indicated by the right-hand noun.
LH: any attributive adjective;
RH: any noun.

7012 N generic type: clause-transitive relation
Explication: the subject of the predicative verb is the actor of the infinitive activity; the adjective expresses an assessment of the subject as actor of the specific activity indicated by the infinitive (e.g. "John was clever to go away"; paraphrase: to go away was clever *of* John).
LH: any form of predicative verb + adjective of type E;
RH: infinitive.

7016 N generic type: determiner//noun
Explication: the adjective anaphorically refers to the 'owner' of the item indicated by the noun.
LH: possessive adjective;
RH: noun or noun phrase.

RECEIVED April, 1969

REFERENCES

1. VON GLASERSFELD, E., PISANI, P. P., BURNS, J., AND NOTARMARCO, B. Automatic English sentence analysis, ILRS T-11 and ILRS T-14. IDAMI Language Research Section, Milan, Italy, 1965-1966.
2. VON GLASERSFELD, E. AND PISANI, P. P. The Multistore System MP-2. Georgia Institute for Research, Athens, Ga., 1968.
3. CECCATO, S. L'Ecole opérationelle et la rupture de la tradition cognitive, *Bulletin de la Société Française de Philosophie*, (Mars-Mai, 1952), Paris, 1953.
4. CECCATO, S., et al. *Mechanical Translation: The Correlational Solution*. Center for Cybernetics, U. Milan, Italy, 1963.
5. CHOMSKY, N. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, Mass., 1965.
6. FILLMORE, C. J. Entailment rules in a semantic theory. The Ohio State U., Columbus, O., 1965.
7. REICHENBACH, H. *Elements of Symbolic Logic*. Free Press, Macmillan, New York, 1966.