I. REL English in Terms of Modern Linguistics

REL, a Rapidly Extensible Language System, is an integrated information system operating in conversational interaction with the computer. It is intended for work with large or small data bases by means of highly individualized languages. The architecture of REL is based on theoretical assumptions about human information dynamics [1], among them the expanding process of conceptualization in working with data, and the idiosyncratic language use of the individual workers. The result of these assumptions is a system which allows the construction of highly individualized languages which are closely knit with the structure of the data and which can be rapidly extended and augmented with new concepts and structures through a facile definitional capability. The REL language processor is designed to accommodate a variety of languages whose structural characteristics may be considerably divergent.

The REL English is one of the languages within the REL system. It is intended to facilitate sophisticated work with computers without the need for mastering programming languages. The structural power of REL English matches the extremely flexible organization of data in ring forms. Extensions of the basic REL English language can be achieved either through defining new concepts and structures in terms of the existing ones or through addition of new rules.

The REL dialect and idiolects

English is our primary mode of verbal communication, therefore everyone has the right to know what someone else means by it. We use the term "English" in its most ordinary sense, i. e. we bear in mind the fact that there really is no one English language. Rather, the term English refers to as many idiolects as there are speakers, these idiolects being grouped into dialects. The REL English is one such dialect. It shares with natural language also the characteristic of being, in its design and functioning, a conglomerate of idiolects, which we call versions. Thompson's design philosophy of REL [2] defines the theoretical basis for the assumption of individual, idiolectal approach to the use of information.

REL English as a formal language

The second basic characteristic is that REL English is a formal language. The characteristics of English as a formal language are discussed in an earlier paper [3]. The central thesis of that paper is that English becomes a formal language when the subject matter which it talks about is limited to material whose interrelationships are specifiable in a limited number of precisely structured categories. It is the type of structuration of the subject matter and not the nature of the subject matter itself that produces the necessary limitations. Natural language encompasses a multitude of formal languages and it is the complexities of the memory structures on which natural language can and does operate that account for the complexities, flexibility and richness of natural language. These latter give rise to the notorious problem of ambiguities in natural language analysis.

Ambiguities

What about ambiguities in REL English? The purpose of REL English grammar is to provide a language facilitating work with computers. It is thus assumed that the language is used for a specific purpose in a specific context. Allowance for ambiguities at the phrase level, with subsequent disambiguition through context, is a powerful mechanism in a language. It is this aspect of ambiguity we wish to include.

Ambiguities, in the general case and in our case, are due to different semantic interpretations (data structuration) arising from different deep structures. Ambiguous constructions are of two main types: (1) those which are structurally ambiguous, e.g., "Boston ships" is ambiguous over all relations existing between "Boston" and "ships" (built in Boston, with home port in Boston, etc.); and (2) those which are semantically ambiguous, e.g., "location of King" if "King" can refer both to Captain King and the destroyer King in the data elements. Ambiguities of the first type can be resolved by the specification of the relation, those of the second type by inclusion of larger context. Chomsky's well-known example of an ambiguous sentence "Flying planes can be dangerous" is of the first type; Katz and Fodor's "bachelor" is of the second type. The purpose of REL sentence analysis is not to find all possible interpretations of ambiguous sentences irrespective of context. Rather, the purpose is maximal disambiguation where such disambiguation is possible in terms of semantic interpretation, providing for the preservation of ambiguities present in memory structures if the syntactic form of the query is ambiguous.

Nature of restrictions

How does our English compare with English as discussed by modern linguists? On the level of surface structure, they are essentially the same. Some more complex transformationally derived strings, such as certain forms of elipsis are not handled as yet. However, most of the common forms are treated in a straightforward manner. Although some constructions which can be formed in natural conversational English are not provided in the basic English package, such deficiencies can to a large extent be overcome by the capability for definitional extension provided by the system.

The level of deep structure presents more problems. As distinct from surface structure, deep structure is that level of syntactic analysis which constitutes the input to semantic analysis, both in Chomsky's [4] terms and ours. What is the nature of this semantic interpretation?

In the general case, little is known. In our case, as in most types of computer analysis, interpretation is in terms of the internal forms of organization of the data in memory. To the extent that the constituents of deep structure can be directly correlated with corresponding structures in the data, semantic analysis, and therefore sentence analysis, can be carried to completion.

It is important to distinguish, in this regard, between two quite distinct though related ways in which language use can be restricted. The first is by the ways in which the data is organized, that is the structural forms used and the interlinkages which are formed for the manipulation of these structures. This type we will call "structural" restrictions. The second is by restrictions of the subject matter, or the universe of discourse; this we will call "discourse" restrictions. When one restricts the universe of discourse to a body of material which is naturally formal or has been formalized, one often tacitly accepts the structural restrictions thus imposed. To the uninitiated, it may appear that it is the discouse limitations and not the implied structural limitations that make the material amenable to machine analysis. However, it is the establishment of relatability between deep structural constituents and data structural forms, rather than discourse restrictions that make computer processing of the semantic component possible. Any content area whose data is organized into these given structural forms can be equally efficiently processed by a system establishing such interrelationships.

The restrictions on REL English are a function of the first type of restrictions, i.e., structural restrictions. Not all deep structures found in natural English are brought out by our analysis, because constituents of these deep structures do not correspond to structural relations in the organization of our data. For instance, "collections of boys" and "boys' collections" are considered synonymous, although they are not in English. Consider: "At the fair, I saw collections of boys." and "At the fair, I saw boys' collections."

Finally, a limitation in reverse, as it were, is the fact that we have emphasized the inclusion of grammatical strings rather than attending to strict insistence on grammaticality.

Organization of the REL English grammar*

The REL English grammar consists of a syntactic component and a semantic component. The syntactic component consists of a

*We are indebted to Norton Greenfeld for many valuable insights and comments as well as for the excellent programming of a large portion of the current English rules.

set of rewrite rules, context-free and general, which build the deep structure Phrase-markers in the form of kernel sentences, and a number of transformational rules. The semantic component consists of semantic transformation rules acting on the memory structures of the data bases. The language processor, described in detail in the accompanying paper [5], builds a sentence analysis Phrase-marker. The sentence is interpreted by successive application of the corresponding semantic transformations in accordance with the parsing. The application of semantic rules to the constituents of a complex phrase produces the interpretation of the phrase or, if the phrase is semantically meaningless, an indication of that to the language processor.

The parts of speech of REL English are given in Figure 1, together with examples. These parts of speech are inclusive terms for syntactic classes (labels on the parsing tree), and semantic categories (memory structures).

Function words, e.g., all, of, what, are distinct from referent words; the former are empty in the sense of not being associated with memory structures. Other aspects of the grammar, namely features, name and relation modification, verbs, clauses, quantifiers and conjunctions, are discussed in detail in the remainder of the paper.

II. REL Data Structures

Structural Power of Language and Structural Power of Data Organization

The case of English as a programming language against other programming languages needs a defense. There are obvious justifications. For a user for whom the computer is only a tool facilitating interaction with his data, English, being his natural mode of language communication, is a medium allowing maximum concentration on the problem itself by minimizing the need for concern with the medium. The flexibility of natural-like language augmented by the extensibility of such a language provides a means for facile manipulation of existing data structures and for new concept formation. We believe that a deeper justification lies in the relationship between language structure and data structuration. There must exist a correspondence between the power of language and the power of data organization. Given the following extreme alternatives:

	1) highly formatted data	2) powerfully structured data
1) formatted language	desirable	undesirable
2) powerfully structured language	undesirable	desirable

the 1) - 1) and 2) - 2) combinations are optimal. The use of a flexible,

rich language for work with highly formatted data introduces too many alternatives and ambiguities which may actually lower efficiency. Conversely, the use of a formatted language for work with powerfully structured data precludes full and efficient exploitation of the organization of the data by imposing a restricted medium of access.

Ring Structure of REL English Data Bases

Since effective communication requires the maximization of information, the structuration of the process must be a compromise between the communicating entities -- in our case, the human user and the computer. Ring structures are a very flexible medium, suitable for the organization of data on which natural language operates and formalized enough to be amenable to manipulation by computer programs. [6]

The semantic component of REL English, mentioned above, consists of the interpretation part of the rules of grammar. These interpretation rules constitute checks on the ring structures. Let us take as our data base the following environment:

1) There are three ships: Maru, Pinta and Nina.
2) The respective locations of these ships are: Boston, Boston and New York.
3) Boston, New York and Chicago are cities.
4) The respective populations of these cities are 3205, 11366

and 6689, in thousands.

Figures 2 and 3 illustrate how ring structures contain this environment. Let us follow these rings. Links from the SHIP ring lead to Maru, Pinta and Nina. The arrow at intersection points on the SHIP ring pointing down and on the Maru, Pinta and Nina pointing up indicate the class-member relationship: Pinta is a member of the class SHIP. This exemplifies ring structures which relate classes of objects and individual objects. The other type of ring structure are binary relations between rings. If we follow now the link from the Pinta ring through the location ring, we reach the Boston ring. The relation of location holds between Pinta and Boston, thus, in predicate notation, we can say: location (Pinta, Boston). The symbols 0, 1 and 2 refer to the order in this relationship.

Number relations, such as population, relate a ring to a number: population (Boston, 3205). They exist as links within the respective rings to which they refer. However, the numerical data itself is in the ring to which it applies. Thus one finds a referent to population on the Boston ring; and associated with the referent to population is the data entry.

The structural methods used to handle the time aspects of time-related data are not discussed in this paper for reasons of space.

Ring structures are also built during analysis of sentences and undone on their completion. Thus in the processing of the sentence:

What are the locations of ships?

the semantic transformation associated with the 'N → R of N' rule will build a scratch ring of all locations of ships, in the particular case at hand, Boston and New York. This scratch ring will be interrelated to the data base in the same manner as permanent rings and thus will participate in further analysis in a straightforward way. Upon completion of the analysis and response to the given sentence, the links to this scratch ring and the scratch ring itself will be deleted from other rings of the data base.

A second type of scratch ring is a number ring, that is a ring all of whose entries are numerical data. For example, the phrase: 'populations of cities' will result in the creation of such a number ring containing the numbers 3205, 11366, 6689. The syntactic result of 'population of cities' has part of speech U, and the number ring feature (discussed below) designates the character of the underlying structure.

Rules such as 'U → average U', or 'U → sum of U' apply to number rings. New data can be entered simply from the console, for example that the Santa Maria is a ship located in Boston:

def: Santa Maria: = name

The Santa Maria is a ship!

The location of the Santa Maria is Boston!

This will result in the creation of a ring for Santa Maria and a linking ring between that ring, location and Boston.

Features and Their Functions

Features constitute an important and powerful mechanism in REL. We use the term in the general Chomsky sense [4], but with some significant modifications. Some aspects of the operation of features from the point of view of the language processor are discussed in [5]. Summarily, the role of features is subcategorization, determination of syntactic constructions and order of syntactic groupings.

(a) There are only two semantic features in the present REL English. The tense character of the verb is a feature assigned to the individual lexical item. The feature distinguishing a number ring from a number is assigned to a data structure which in a sense is also treated as a lexical item in sentence analysis.

(b) Morphemic features subcategorize syntactically parts of speech. These include the Nominative/Possessive and Singular/Plural subcategorizations of Nouns and Relations; and the Singular and Participle subcategorizations of Verbs. Morphemic features often function as precedence markers as well. Thus checks on Nom./Poss. and Sing./Plural features force the following parsing:

(((small (computer s))') (word size))

excluding such parsings as:

(small (((computer s)') (word size)))

(c) The passive voice feature of verbs is set by the verb "to be" used as an auxilary. Once set, it signals other routines to appropriately handle subject and object, essentially keying the passive transformation. The negation feature on the verb is set by "not" and its contracted forms.

(d) Precedence features. The remaining features determine the applicability of rules and the order of their application to form syntactic groupings. On N and R they are:

determiner -- which prevents further modification

e.g. the the ball, big the ball, are excluded.

phrasal -- which prevents the application of certain morphemic rules to a syntactically modified phrase.

pre-modification and post-modification -- which cause the following constituent analysis:

e.g. (the (dentist (son of (Sally Smith))))

not: (((the dentist) son) of (Sally Smith))

They also are used to maintain parallel construction in conjunctive phrases.

e.g. ((John's son) and (Mary's daughter))

not: John's (son and (Mary's daughter))

while: (Mary's (uncles and aunts))

not: ((Mary's uncles) and aunts)

On the V phrases, right modification and subject features play a similar role. Auxilaries must go on before right modification; right modification must be completed before the subject can go on.

e.g. (John (((will marry) Joan) before 1970))

not: ((John (will (marry Joan))) before 1970)

The importance of precedence features is that they sharply curtail redundant parsings. Thus the rules:

N → Ns	N → NN
N → N's	N → the N

give seven parsings of the phrase 'the boy's dogs'. The feature checks eliminate all but one, namely: (the ((boy's) (dogs))). Since these feature checks are made by the parser at the earliest possible moment, they provide a very efficient and effective control on redundancy.

III. Building of Linguistic Structures

Name Modification

A "name" refers to a class or the individual members of a class in the ring structured data base, a "relation" to a binary predicate.

Name modification constructions consist of a name as head and one or more modifiers. Modifiers are: determiners, relations, and other names.

The indefinite and definite articles comprise the first class. Their functions are complex, and an adequate discussion would go beyond the scope of this paper.

Relations can function as pre-modifiers and post-modifiers. Accordingly, there are two different syntactic constructions, which we treat in REL as synonymous: "location of John" and "John's location".

Figures 4 and 5 illustrate the application of such rules.

Relation Modification

We distinguish between two types of relations, primitive and complex. A primitive R directly designates an associated ring structure. A complex R designates a list structure consisting of constructions which build a complex relationship out of primitive ones. The process of building this list structure is referred to as relation modification. Figure 6 shows the possible constructions, together with grammar rules for relation modification. Semantic transformations for name modification rules recursively resolve this list structure. For example, in the analysis of the phrase:

John's male child's location

"Male child's location" gives rise to the complex R:

composition [modification [male, child], location].

Then the name modification rule is applied to: "John's R", whose semantic transformation expands the complex R, developing the location of each "male (child of John)".

Relation modification plays an important role in definitions of complex concepts. It is employed widely in elevating languages beyond the base level of primitive relationships. This can be exemplified in family relationship situations. For instance, given the "parent" and "identity" relations as primitive, the following series of definitions introduces the relation "uncle":

def:child:converse of parent

def:son:male child

def:sibling:child of parent and not identity

def:brother:male sibling

def:uncle:brother of parent

Relation modification also plays an important role in building complex definitions involving number relations such as area, GNP, salary, etc. Consider the definition:

def:per capita "GNP":"GNP"/population

Complex relations of this sort function in such expressions as "Average per capita area of nations".

The Verb

We are accustomed to the traditional definition saying that a verb denotes an action or a state: an action is performed by an actor (subject) on an object, a state is a momentarily or permanently frozen action between subjects and objects. Thus, an action, in this inclusive sense, is characterized by (1) the aspect of beginning, ending, duration or momentariness; (2) by its situation in time, and (3) by referring to subjects and objects. Two groups of verbs may be distinguished: those referring to a relation between subjects and objects, and those which establish a connection between them. The relation expressed by a verb constitutes (and is here referred to as) its 'predicate'; such verbs are called relation verbs. Relation verbs also express temporal aspects of a relation. Typical relation verbs are "arrive" and "leave"; both refer to the relation of 'location'; "arrive" refers to the beginning of the existence of this relation, and "leave" to its ending. For instance, "John left Boston" means that the relation of 'location' existing between John and Boston came to an end. Verbs which express a connection between subjects and objects are referred to as copulas, e.g. "is" in "John is a boy". The copula itself constitutes the predicate.

1. Relation Verbs

Each of the four following sentences contains, in surface structure, a different verb.

(1) John arrived in Boston.

(ii) John left Boston.

(iii) John lived in Boston.

(iv) John is residing in Boston.

The underlying structure of these sentences is identical: 'location (John, Boston)', except for the temporal aspect of this relation: beginning in (i), ending in (ii), momentariness in (iii) and duration in (iv). In REL English, the temporal aspect is denoted by the 'tense character' feature.

Verbs are introduced through language extension. They are defined in terms of a relation and a tense character. The relation must denote an already existing ring structure. Thus, given the relation of "location", the verb "arrive" is defined by:

def:arrive:verb (location, 1).

A verb is internally represented as a "verb table", with part of speech V. In the verb table for "arrive", 'location' is entered as predicate and '1' as tense character, indicating beginning.

Initially, the verb table is assigned the present tense, or 'nowness'. This can be subsequently modified by verb morphology, auxiliary verbs, and tense modifiers (temporal phrases or clauses).

2. Copula verbs

Copula verbs have a verb table which has a copula predicate and (at present) no tense character. The copulas are: "is", "are", "was", "were", and their contracted negatives. For example, the rule 'C → wasn't' will result in the creation of a verb table with a copula predicate, time equal to past, and the feature of negation.

3. Verb table modification

The elements of a verb table correspond to the elements of a kernel clause (i.e. one with a single deep structure Phrase-marker). They are: subject, predicate, object, tense character, and times t_1, t_2. The subjects and objects of the kernel may be N (names) and U (numbers). The predicate is a relation. The types of the tense character are discussed above. The times on a verb table indicate a point or interval within the span bounded by past (0) and future (∞). The t_1 and t_2 either both specify a single point or bind an interval of time in which the event indicated by the clause takes place.

REL verbs are marked by three regular inflectional morphemes: past tense, past participle, and 3rd person singular. The function of inflectional morphemes and auxiliary verbs is two-fold: modification of the original time in the verb table and setting of syntactic features. For example, the past tense morpheme and the auxiliary "did" modify t_1 to be past, thus establishing the time interval '0 to now' (i.e. $t_1 = 0$, $t_2 = now$); the auxiliary "will" modifies t_2 to be future, thus establishing $t_1 = now$, $t_2 = \infty$; 3rd person singular and auxiliaries "has" and "does" set the singular feature.

Time modifiers (M) further modify the verb table. They originate from prepositional time phrases, e.g. "before 1960", and subordinate clauses, e.g. "before John arrived in Boston". The times of the original verb table (or as modified through the past morpheme and/or auxiliaries) are further modified so as to be in accord with time modifiers. Thus, in the sentence "Will John arrive before 1970?" the verb table for "arrive" has t_1, t_2 = now; modification by "will" changes t_2 to '∞'; modification by "before 1970" changes t_2 to '1970', thus resulting in the intersected time interval 'between now and 1970'.

Subjects and objects are inserted in the verb table upon application of the rules:

(i) $V \rightarrow N\ V$ (rule putting subject on)

(ii) $V \rightarrow V\ N$ (rule putting object on)

(iii) $V \rightarrow V$ by N.

These rules also check whether the passive feature is on (set by copulas forming passives), and if it is, rule (i) converts the N into an object; rule (ii) converts the N into a subject; rule (iii) applies only if the passive feature is on and converts the N into a subject. Number subjects and objects do not participate in the passive transformation.

Negation is handled by a feature on the verb table. It is set by rules processing "not" with copulas, both in isolation and contracted, and rules processing "not" with verbs. They differ syntactically in the position of "not". If the negative feature is already set by a previous rule, the result is positive (i. e. double negatives turn into a positive).

Let us consider an example of verb table accumulation (figure 7). The development of the verb table shows the successive modification due to the verb rules which have applied. The parsings reflect the use of features which eliminate some possible parsings which are incorrect or irrelevant to the analysis, and determine the order of parsing.

Clause Processing

The accumulated verb table constitutes the input to the clause processing routine. The verb table may be either complete, i. e. have all its positions filled, or else have subjects or objects missing. The clause processor completes the verb table if necessary, and returns it together with a list of times for which the clause holds, and another list of times for which it does not hold. The results of the clause processor are subsequently used by the sentence rules to produce output, or by the subordinate clause rules.

Sentences are clauses plus sentence delimiters -- in REL English, these are initial capital letters, and terminal question mark or exclamation mark. Accordingly, we distinguish two types of sentences, questions which interrogate the data, and commands which modify it.

The rules for handling questions are: (S stands for sentence, K for capital letter)

(i)	S → K V?	Did Stan marry Jill before 1950?
(ii)	S → K what/who V?	Who are the Smiths' children?
(iii)	S → K M?	When did John arrive?

Rule (i) results in a yes/no output depending on whether the relation holds at the specified time. Suppose the verb table is given us

predicate	spouse
subject	Stan
object	Jill
tense character	begin
t_1	0
t_2	1950

Suppose also the data show that Stan married Jill in 1948. Since 1948 is within the time interval '0 to 1950', the answer will be affirmative If the input is a negative question, we use the time for which the relation does not hold.

The output of rule (ii) is one or more Ns. These are supplied by the clause processing routine as either subjects or objects. In example (ii), subjects are supplied.

The output of rule (iii) is a time (or time list) at which the relation indicated in the verb table holds.

The output may be ambiguous. For example, "Did Smith live in New York?" would result in both "yes" and "no" as ambiguous output if "Smith" referred to one Smith who did and another who did not live in New York. "Who is Stan's sister?" would have ambiguous answers if Stan had more than one sister.

The output is a Vacuous Description if there is no valid semantic analysis of the query. For example in the sentence, "What is the income of the present king of France?" the phrase "present king of France" is a vacuous description in terms of our data if the data contains factual information on present-day France. This aspect of the analysis is not explicated here for lack of space.

The second type of sentences, commands, are used for adding or deleting data. The rules and corresponding examples are:

(i) $S \rightarrow K\ V!$	John is a boy!
	Sally lived in Boston after 1950!
(ii) $S \rightarrow K\ R$ of $V!$	The location of Sally was Boston after 1950!
	The income of John is 10000!

The data is deleted if the verb table has the negative feature set. There are two restrictions on the above rules: the subject of the verb table in (i) must not be modified; and the predicate of the verb table in (ii) must be a copula.

The structure of the input sentence determines the structural relations to be established between items in the data.

Subordinate clauses modify some item in another clause, or another clause as a whole. In REL English, clauses of the first type are always relative clauses; they are introduced by the pronouns "who", "which", "that", "whom" and "whose". Clauses modifying other clauses are temporal clauses introduced by "before", "after" and "when", and result in time modification.

Examples of relative clauses:

(i)	N → N who V	Did the boy who left Boston marry Jill?
(ii)	N → N that V	Was Boston the city that John left?
(iii)	N → N whose R V	Did the boy whose father left Boston marry Jill?

Verb tables representing relative clauses are completed using the noun which the clause modifies. Thus in (i) the N is used as the subject of the verb table, in (ii) either as subject or object. In (iii) the N for which the relation R holds is used as subject or object.

Parallel to the above rules are rules with a comma preceding the relative pronoun. However, the rules with commas apply only to post-modified N's and the comma plays a disambiguating function. For instance, "parents of boys who left Boston" can be ambiguous; in our English, the relative clause refers to "boys", while in "parents of boys, who left Boston" the relative clause refers to "parents".

Examples of temporal clauses:

(i)	M → before V	Did John marry Jill before Jill left Boston?
(ii)	M → when V	What was John's income when Stan lived in Boston?

The output of rule (i) is a time modifier with the time interval $t_1 = 0$, t_2 = date specified by the subordinate clause. In the subsequent analysis these times are compared with the times on the main clause in the same way as time modifiers such as "before 1960." For instance, let us assume that John married Jill in 1950 and that Jill left Boston in 1955. The output of rule (i) will now be the interval '0 to 1955' and the times to be intersected are '0 to 1955' and 1950:

```
0 ------+---------------+------.
       1950            1955
```

The result is, obviously, a "yes" answer. The output of rule (ii) is a tense modifier with the time interval $t_1 = t_2 =$ date specified by the subordinate clause.

<u>Quantifiers</u>

By "quantifiers" we refer to "all", "some", "how many", "what", "each" and "no". Phrases containing such words are of particular importance since they often are used in referring to aggregates and more abstract concepts.

Some examples are:

Did some Smith who lived in New York move to Boston?

How many Smiths have an income greater than 12000?

What was the average between 1950 and 1960 of the incomes of each Smith?

In such sentences, a class of objects is considered. The sentence constitutes a condition on each element of this class, the condition depending on the particular quantifier. Consider in detail the example: "Is Boston the location of all men?" To answer this question, each man must be considered in turn. We apply the term generation to this kind of a process. As each man is generated in turn, say man_i, the sentence "Is Boston the location of man_i?" must be processed. If the answer to all cases is "yes", then the original question is answered affirmatively; otherwise, it is answered negatively. In the case of the "some" generator, the answer would be affirmative if at least one of the men lives in Boston. For the question "What men live in Boston?", the answer is the list of men for whom the clause "man_i lives in Boston" holds.

Several generators can be nested in a sentence. An example is: "Each Smith lived in what cities?" In general, the nesting of generators poses difficult and challenging problems. Although we ıave been able to satisfactorily deal with these problems, an ıdequate treatment must be deferred to a separate publication because ıf their complexity.

Conjunctions

Conjunction of R's was discussed above under relation modifiation. Conjunction of N's and V's set up generators according to the ollowing rules:

$$N_{GE} \rightarrow N\ J\ N$$

$$N_{GE}J \rightarrow N,\ NJ$$

$$V_{GE} \rightarrow V\ JN$$

$$V_{GE}J \rightarrow V,\ VJ$$

here the GE subscript indicates a generated phrase. If the J-phrase s "and", the generator is an "all" generator; if the J-phrase is "or", ıe generator is a "some" generator. An example is given in figure 8.

IV. A Few Inclusive Examples

```
>What was the average between 1964 and now of the incomes of
each woman who has not married?
Sue Smith                    0
Sally Smith                  10,892.3
>Each Jones lived in what city when Jill Jones arrived
in Los Angeles?
Mary Martin Jones          Boston
Jack Jones                 Boston
Jill Jones Smith           Los Angeles
John Jones                 Boston
>Did Stan Smith marry Jill Jones and live in Los Angeles?
Yes
>What were the incomes of Steve Smith's parents after Steve
Smith arrived?
Jill Jones Smith     4,000 August up to December, 1962
                         0 from 1963 up to June 1969
Stan Smith           8,000 August up to December, 1962
                     9,000 in 1963
                    10,000 from 1964 up to 1967
                    11,000 from 1967 up to June 1969
>When was the number of Smiths equal to 5?
AMBIGUOUS OUTPUT:
 (1)    August 1938 to September 1956
 (2)    September 1958 to May 1960
>When did each Jones live in each city?
John Jones                 Boston                  1930 to June 1969
Jill Jones Smith           New York                September 1955 to Apr
                           Los Angeles             June 1961 to June 196
                           Boston                  July 1937 to August 1
Jack Jones                 Boston                  1935 to June 1969
Mary Martin Jones          Boston                  1930 to June 1969
>def:in the "1940"s:between "1940" and 10 years after "1940"
DEFINED.
>What was Stan Smith's income in the 1960s?
     6,000 in 1960
     8,000 from 1961 up to 1963
     9,000 in 1963
    10,000 from 1964 up to 1967
    11,000 from 1967 up to June 1969
>_
```

REFERENCES

[1] Thompson, F. B., "The Dynamics of Information," Virginia Quarterly Review (in press).

[2] Thompson, F. B., et al., "REL: A Rapidly Extensible Language System," Proc. National ACM Conf., August 1969 (in press).

[3] Thompson, F. B., "English for the Computer," Proc. AFIPS Fall Joint Comp. Conf., 29 (1966), 349-356.

[4] Chomsky, N., Aspects of the Theory of Syntax, MIT Press, 1965.

[5] Lockemann, P. C., Thompson, F. B., REL: A Rapidly Extensible System, I: The REL Language Processor.

[6] Craig, J. A., Berezner, S. C., Carney, H. C., Longyear, C. R., "DEACON: Direct English Access and Control," Proc. AFIPS Fall Joint Comp. Conf., 29 (1966), 365-380.

Symbol	Name	Examples
N	name	boy, John, the location of John
R	relation	location, father, age
V	verb table	arrive, has lived
C	copula	is, was
J	conjunction	and, or
T	time	January 1960
M	time modifier	before January 1960 between now and 1970
U	number	52, 61.34
S	sentence	John is a boy! Is John a boy?

Figure 1 Parts of Speech

Ring Structured Data

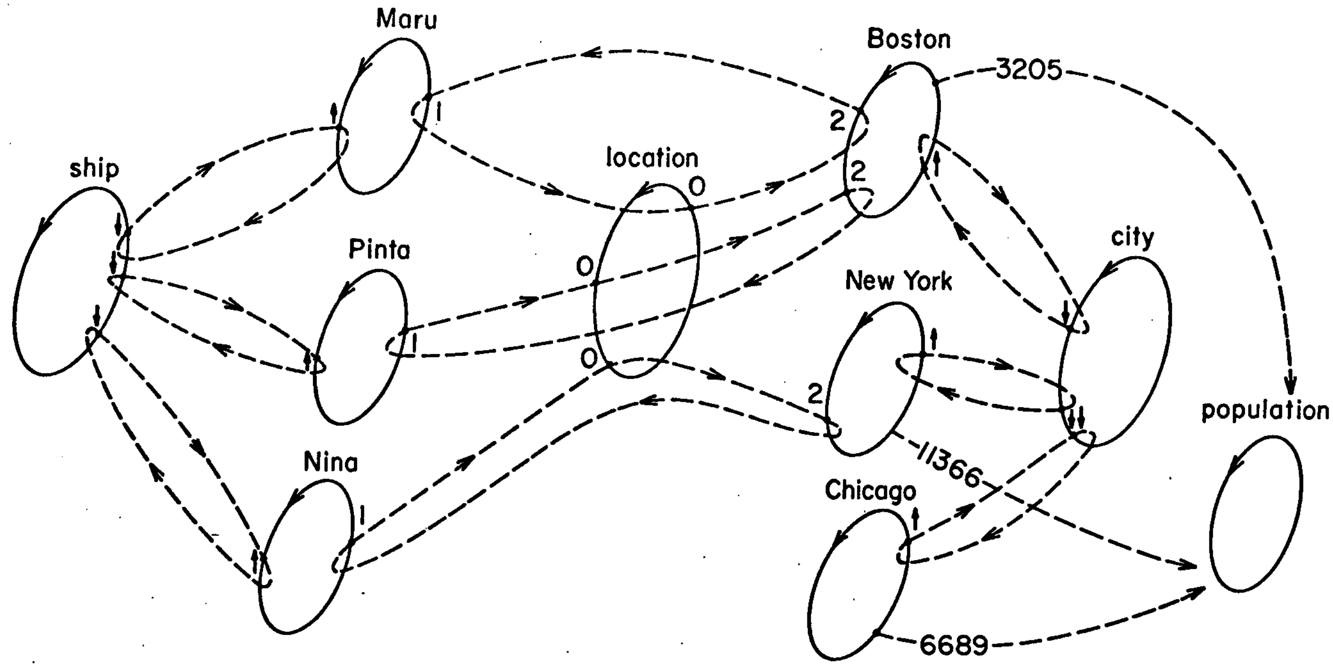


Figure 2

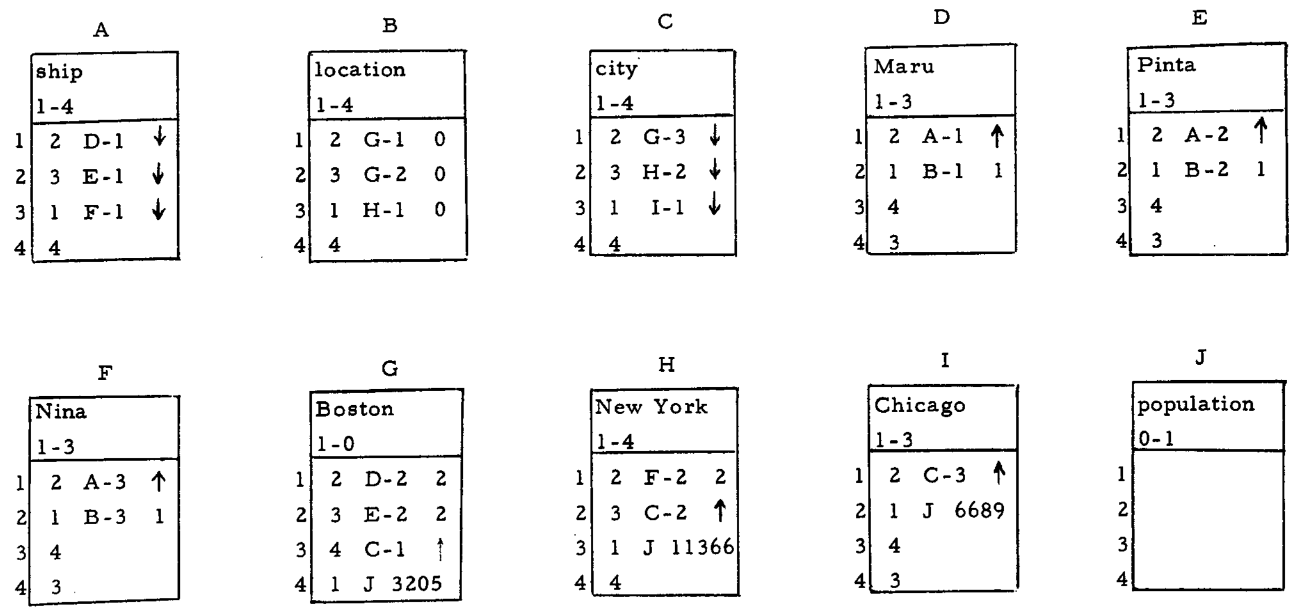


Figure 3 Schemantic Page Layout of Ring Structured Data

type of constituents		example	output N
R	N		
singular	singular	location of John	scratch ring of all locations of John, singular, phrasal, postmodified
plural	singular	locations of John	scratch ring of all locations of John, plural, phrasal, postmodified
singular	plural	location of boys	scratch ring of all locations common to all boys, singular, phrasal, postmodified
plural	plural	locations of boys	scratch ring of all locations of all boys, plural, phrasal, postmodified

*This case raises an interesting question: What is meant by location "common" to all boys--common at all times, or common no matter what time?

Figure 4 Cases for rule N → R of N

type of constituents		example	output N
N (singular only)	N		
class	singular individual	scout Smith	Smith who is a scout if there is only one. List of such Smiths, marked as ambiguity, if there is more than one. singular, phrasal, premodified
class	singular class	boy scout	Scratch ring of all boy scouts. singular, phrasal, premodified
class	plural class	boy scouts	Scratch ring of all boy scouts. plural, phrasal, premodified
individual	singular class	Boston ship	For each relation that interrelates Boston with some ship, a scratch ring for all such ships. Ambiguous over all such relations. singular, phrasal, premodified
individual	plural class	Boston ships	Same as above, but plural

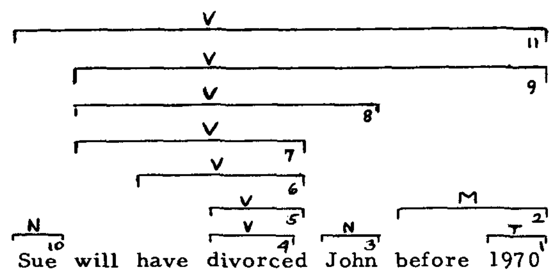
Figure 5 Cases of Rule $N \rightarrow NN$

composition	R → RR	son's location
conjunction	R → RJR	uncles and aunts
disjunction		uncles or aunts
modification	R → NR	boy friend
relative negation	R → R but not R	uncles but not aunts
converse	R → converse of R	converse of parent

(for number relations)

sum	R → R + R	direct salaries plus overhead
difference	R → R - R	
product	R → R * R	pay rate * hours worked
quotient	R → R / R	GNP/population

Figure 6 Complex Relation Constructions



Verb Table	4	5	6	7	8	9	11
predicate	spouse	spouse	spouse	spouse	spouse	spouse	spouse
subject	-	-	-	-	-	-	Sue
object	-	-	-	-	John	John	John
tense character	end	end	end	end	end	end	end
t_1	now	0	0	0	0	0	0
t_2	now	now	now	∞	∞	1970	1970
(participle feature)	off	on	off	off	off	off	off

Figure 7 Verb Table Accumulation

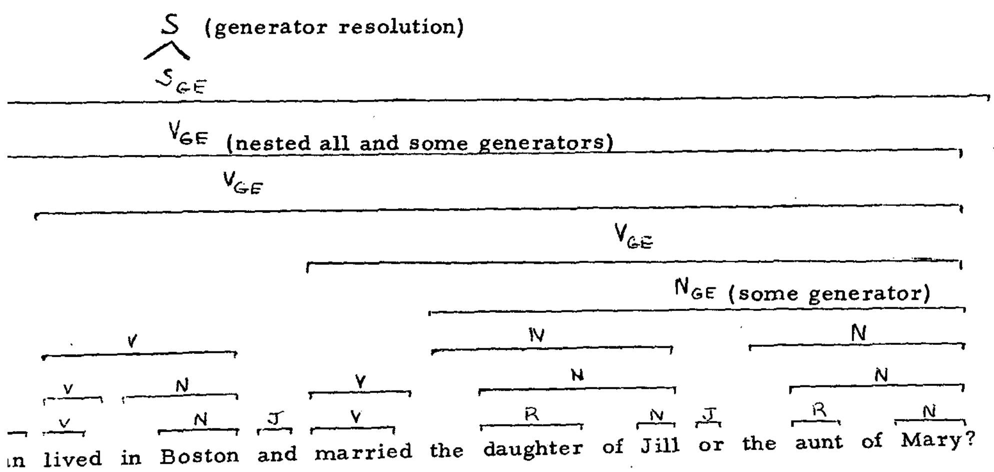


Figure 8 Example of Conjunctions