H. Lehmann

Interpretation of Natural Language in an Information System

Abstract: This paper discusses some of the linguistic problems encountered during the development of the User Specialty Languages (USL) system, an information system that accepts a subset of German or English as input for query, analysis, and updating of data. The system is regarded as a model for portions of natural language that are relevant to interactions with a data base. The model provides insight into the functioning of language and the linguistic behavior of users who must communicate with a machine in order to obtain information. The aim of application independence made it necessary to approach many problems from a different angle than in most comparable systems. Rather than a full treatment of the linguistic capacity of the system, details of phenomena such as time handling, coordination, quantification, and possessive pronouns are presented. The solutions that have been implemented are described, and open questions are pointed out.

Introduction

During construction of the User Specialty Languages (USL) system, a number of linguistic problems were encountered; these had not been treated with sufficient detail in the literature to permit ready implementation of solutions. The solutions found for the USL system in these cases are felt to be of interest also outside the environment of data base interaction via natural language.

The USL system was created to provide users with a tool for accessing and analyzing data without having to become expert in electronic data processing. It was assumed, however, that the user would be a professional knowledgeable in his field, not the casual user as described by Codd [1]; and, therefore, that the system should allow him to express himself in the terminology he was used to. The system was to be application-independent: no features dependent upon subject matter should be present in the language processing part.

An independent data base management system (DBMS) was required for the construction of the USL system, making it possible to benefit from the work done in data base research. To maintain a well-defined interface, input sentences were translated into the formal data manipulation language of the DBMS (a similar approach was also taken by Mylopoulos et al. [2], Sacerdoti [3], Waltz et al. [4], and Sibuya et al. [5]).

A revised version of Kay's parser [6] was used for syntactic analysis. The method of interpretation used in the REL system [7–9] was taken as a point of departure, but this method was augmented to more adequately handle coordination, quantifiers, and possessive pronouns. Hence, the principal work required for the design and implementation of the present system involved constructing a grammar for German that could be recognized by the parser and developing suitable interpretation routines that would perform the mapping from German to the data manipulation language.

The USL system is comparable to question-answering systems, a number of which have been developed during the past fifteen years. Surveys of such systems can be found in [10–12], and comparisons to USL are given in [13], while a comparison between the systems TQA (formerly REQUEST) [14, 15] and LSNLIS [16] is given in [17].

An overview of the USL system is given and similarities to other existing systems are indicated. Thereafter, the semantic concepts underlying the system are introduced in order to provide a basis for the discussions that follow. They concentrate upon four kinds of linguistic structures considered essential for a question-answering system. Without these structures, important sets of queries cannot be formulated.

Temporal expressions A comparison with Bruce's CHRONOS system [18] shows that despite his sophisticated model of time, many relatively simple but essential aspects have not been addressed. Two problems are dis-

cussed: proper treatment of vagueness in temporal references, and conversion of deictic references (e.g., "last week") to actual dates.

Coordinated sentences and phrases These are among the most difficult structures in natural languages. Their treatment in this paper is not considered exhaustive, although it is hoped that a contribution has been made to the clearer understanding of the coordination phenomena. The possible interpretations of such structures are discussed, and criteria are given that determine the respective interpretations.

Quantifiers There are still many unresolved questions concerning quantifiers; the treatment here concentrates on a discussion of the scope of quantifiers and on their interplay with particles of negation.

Contextual reference This problem is addressed in the USL system only with respect to possessive pronouns. Criteria for their reference are discussed, and reasons given why a completely formal treatment is not possible at the present time. The solution found for the USL system is presented and justified.

The examples used in the discussions are in English unless German and English differ in their behavior, in which case German examples with English glosses are given.

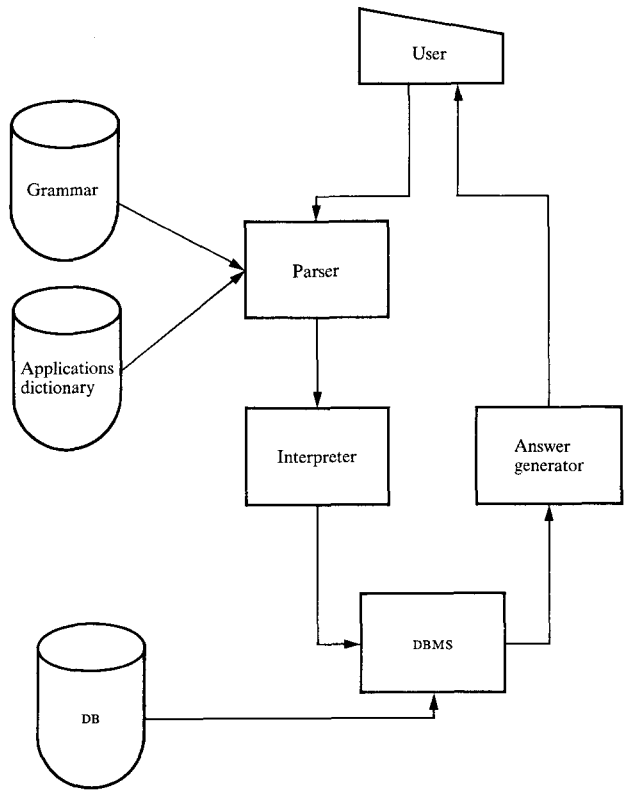

Figure 1 General system design of the USL system.

System overview

The USL system (the general design is shown in Fig. 1) is constructed around the relational data base management system PRTV [19], is coded in PL/I, and runs under VM/CMS in a 2500-Kbyte virtual machine. The system uses a revised form of Kay's parser [6, 20] and a German grammar (comprising some 800 rules in a modified Backus-Naur format) that was developed for the system. Each rule specifies both a syntactic configuration as a condition for its application and one or more categories that replace the original configuration after the rule has been applied. Each rule also contains reference to an interpretation routine, of which there are some 70 in the system [21]. The German grammar was later taken as a basis for the construction of English, Dutch, and Spanish grammars with the same interpretation routines as the German. A dictionary contains all relevant function words of the language whose meanings are independent of particular applications (prepositions, conjunctions, "to be," "to have," names of months, days of the week, etc.). Attached is an application-dependent dictionary containing all those words used to refer to relations (or tables, as explained in the section on semantic concepts). Numbers and words used to refer to objects within relations are not defined, but are recognized instead by so-called variable token rules (where patterns of letter strings are specified, and categories are assigned to strings conforming to the patterns). Morphological endings are recognized by syntax rules; secondary stem forms are entered separately into the dictionary. A detailed description of the German grammar can be found in [22–25].

The clear distinction between function words (general vocabulary) and application-dependent vocabulary has several consequences. One is that the vocabulary must be easy to define: the user should require no specific linguistic knowledge or education in logic. This limits the amount of information about each word that can be used in the linguistic analysis of input sentences. Hence the question arises whether it is possible to achieve a system that can handle a sufficiently comfortable subset of a language with such limited information at its disposal. This question should be answered by the evaluation of the USL system that is being conducted with several different applications.

Another important consequence of application independence is that the meanings of syntactic constructions must be understood regardless of the content-bearing words in these constructions. This distinguishes the USL approach from other approaches to language processing systems in which a depth-first analysis of a very limited subject domain is attempted (see for example [26–28] and the criticism of [28] given in [29]). These approaches were

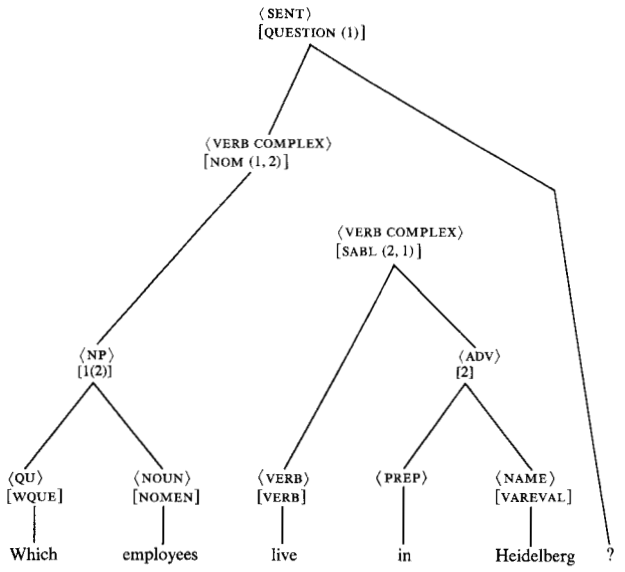

Figure 2 Result of parsing sentence (1).

considered unacceptable for our purpose because methods for generalizing them are not known. Thus extensive empirical investigations into the use and interrelation of expressions and concepts of every new domain (subject matter) would have been required before a system could actually be used for asking questions. It was felt that such adaptations to new domains would hardly be practical since the query language generally represents only part of the total problem to be solved.

Analysis of input sentences

Analysis in the USL system is begun by making each letter of a given input string a node of an initial parse tree. Dictionary entries, variable token rules, and grammar rules are then applied by the parser, which works from left to right, in a bottom-up fashion, and sets up in parallel all syntax trees that fit a given input sentence. Parsing produces one or more trees whose structures still reflect the surface structure of the input sentence. Each node contains the name of an interpretation routine, and they are called consecutively in such a way that each routine takes the outputs of its predecessors as input. This process is best illustrated by the following example:

Which employees live in Heidelberg? (1)

The result of the parsing of (1) is shown in Fig. 2. The calling sequence of the routines indicated at each node can be represented in the following structure:

```
QUESTION(NOM(WQUE(NOMEN('EMPLOYEE')),
                SABL(VAREVAL('HEIDELBERG'),
                        VERB('LIVE')))) (2)
```

Up to this point, the USL language processing method resembles that of the REL system [7–9], except that the USL grammars cover more structures.

The result of the execution of the interpretation routines is an intermediate tree structure that no longer reflects the surface structure, but is more data base oriented. This structure is input to a recursive routine to produce a data base language statement which is passed to the DBMS for execution. The reason for introducing the intermediate tree is that coordination, quantification, and possessive pronouns can be dealt with much more adequately than if data base language expressions were generated directly from the parse trees. The intermediate tree for (1) is

```
R: LIVE
  A(LA  ): 'HEIDELBERG'
  A(NOM): R: EMPLOYEE                    (3)
```

where R stands for *Relation*, A stands for *Argument*, and the symbols in parentheses indicate columns of relations to which the information following the colons refers (see also the section on semantic concepts). Any node of type R may have as many arguments (nodes of type A) as necessary [none in the case of EMPLOYEE in (1)]. Additional information is kept with each node, e.g., the fact that "which" preceded "employee," interpreted as a request to output column NOM of EMPLOYEE. Application of the recursive routine yields

```
((LIVE;C2='HEIDELBERG')*EMPLOYEE;C1=C3)%C1   (4)
```

In the case of coordinate phrases, the intermediate tree looks more complicated:

What are the addresses of Brown and Smith?

```
R: BE
  A(NOM): R: ADDRESS
            A(VON): 'BROWN'  A(VON): 'SMITH'
  A(NOM): ?                              (5)
```

The information kept with nodes of type R includes relation name, comparison operator, quantifier, negation, presence of "which" or "whose" (for nouns), noun with complement (genitive attribute / PP / adverbial), and type of verb coordination (none / and / or). The information kept with nodes of type A includes type of argument (relation / proper name), role name, type of noun coordination, coordination with genitive attribute, definiteness of quantifier, number, and information on possessive pronouns (reference and type of concord). This information is recorded, because it is generated at parsing time and is needed only later for the process of translation to the data base language.

Semantic concepts

In the context of an information system, the meaning of sentences must be correlated with the contents of data files, which themselves can be viewed as tables consisting of rows and columns. Such tables are also called *relations* if they do not contain duplicate rows. Words can then represent names of relations, columns of relations, individual items, or operations on relations (e.g., "insert," "delete," and "display"). Thus, a question such as (1) can be taken to refer to a relation LIVE containing the two columns PERSON and LOCATION, and to a relation EMPLOYEE containing the names of employees. An answer to the question can then be found by going through these relations to check for all people whose LOCATION is HEIDELBERG, and then whether the people are also employees. Operations of this sort can be conveniently performed by a relational data base management system. The remaining task for the interpretation of natural language consists in systematically mapping words, phrases, and expressions to relation names, item names, and operations on them.

It was intended that the USL system should understand those aspects of language that refer to the context of a data base. Since, in a sense, a data base is a model of a section of the world, it was necessary to develop a simplified but general model of the world. Our model consists of three kinds of entities: *objects*, *relations*, and *states*; relations are sets of n-tuples of objects, n being fixed for every relation.

Let the pair $\langle U, R\rangle$, where U is a set of objects and R is a set of relations, be called the *semantic base* S of the world model. The model can be refined by categorizing the objects of U: A set of *domains* D is introduced, where D is defined as a subset of the powerset of U. A set of *standard domains* is defined for the USL system. Only the most general domains are used, since a calculus of domains does not exist in our current data base management system. Instead, one-place relations are used to classify the objects in the universe of discourse. The standard domains are: ZAHL (number), WORT (word, character string), DATUM (date, time of day), and CODE (numeric code).

The notion of relation can be refined by naming the columns of each relation. These names we call *roles*. A set of roles Ro can be defined from which every relation must draw its roles (the ones allowed in the USL system will be listed below).

The semantic model is conceived as a dynamic structure: It can be imagined to consist of states that may differ with respect to U, R, D, and Ro. This concept of state is very similar to Carnap's *state descriptions* [30], or to the concept of *Zustand* in [31].

Figure 3 shows the relationships among language, the world, and models for the world. Natural language reflects in its structure a common-sense view of the world, here called the *linguistic world view*, where concepts such as *thing*, *property*, and *event* exist. This world view varies with speaker or speech community (for a detailed discussion, see [31]).

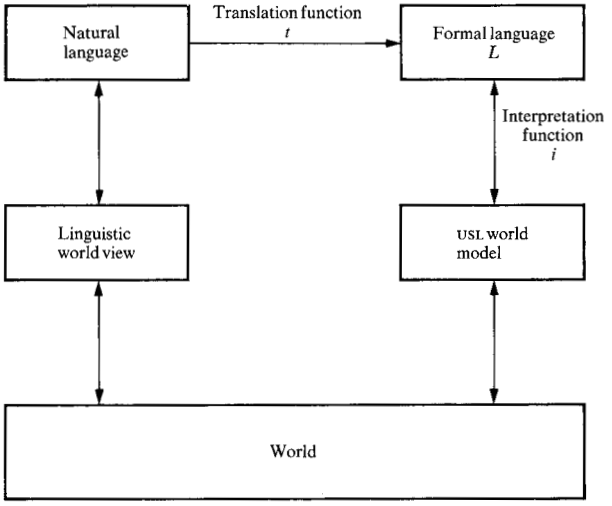

Figure 3 Relationships among language, the world, and models for the world.

The language L also is very simple. It is just adequate to handle everything in the world model and contains a variant of first order predicate calculus and operators for state transitions. A formal definition of L is given in [32].

In the USL system this simple model is used to interpret natural language; thus, not everything that can be expressed in natural language can be interpreted by the system.

Theoretically, natural language can be interpreted in a USL-like system either by direct interpretation of the linguistic structures in terms of the model, or by translation of the linguistic structures into structures of a formal language L by means of a translation function t (see Fig. 3). The latter method was chosen principally because it has the capability of formulating a clear and clean interface to a data base system (one that already performs the interpretation function i of Fig. 3). The same choice was made by Montague [33] for theoretical reasons.

Concepts are expressed by words or phrases in natural language, and can be represented in the model by relations, objects, or state transitions. The way in which a particular concept is represented depends on word class, utilization of the concept, taste, etc. The word classes referred to in this context are *noun*, *adjective*, and *verb*. Nouns can be subdivided into proper names and common nouns. Proper names will generally be represented as names of objects, verbs as names of relations.

Common nouns and adjectives can be used to represent names of objects or relations. In theory, they might also

SUPPLIER	
NOM—NAME	VON—PRODUCT
JONES INC. SMITH & BELL GML	BOLTS NUTS SCREWS

Figure 4 Common nouns associated with role names.

be associated with domain or role names. In the relation shown in Fig. 4, SUPPLIER, PRODUCT, and BOLTS are all common nouns. Note that with an organization such as that shown in the example, the question "Are bolts a product of Jones Inc.?" cannot be asked, unless either 1) PRODUCT is also the name of a relation, or 2) a link from PRODUCT to SUPPLIER exists somewhere.

In the USL system the choice was made to have words refer to object or relation names, but not to domain names. To permit reference to various columns within a relation, *standard role names* were introduced. These role names were defined with respect to (surface) cases governed by a verb, prepositions governed by a verb or noun, and types of adverbials. Reference to deep cases as, e.g., in [2] was not chosen, since it is not clear how users with no background in linguistics would be able to define the proper mappings from surface to deep cases. The standard role names are as follows:

NOM	nominative case, set of objects referred to by noun or adjective
ACC	accusative case
DAT	dative case
GEN	genitive case
VON	genitive attribute
LA	place
LO	origin
LG	goal
LD	distance
LP	path
TA	point in time, date
TO	beginning
TG	end
TD	time interval
⟨preposition⟩	preposition governed by noun or verb

The introduction of standard role names provides more flexibility in the formulation of questions, since relationships expressed by syntactic constructions can be utilized.

An important restriction of the semantic model is that one cannot express statements about statements. It is also not possible to express intensional contexts, a problem recently investigated principally by logicians (see, e.g., [33], [34]); such phenomena will probably become important for future research.

Temporal expressions

Events are associated with dates and time intervals in many applications, but the manner in which dates are generally stored in computers today is not always suitable. Several different conventions are used: e.g., 75073 (73rd day of 1975), 750314, 03/14/75, and 14/03/75. Of the forms given, the first is best suited for calculations on intervals, since ordinary addition and subtraction can be used as long as the year stays the same. The other representations show better what month is referred to, but they are not really good for doing calculations on intervals. The last two examples refer to the same date, according to American or European conventions, respectively. Since representations suitable for the calculation of intervals are not intelligible unless converted, it is obvious that no ideal representation exists for a user-friendly system. The only practical solution involves displaying the dates so that they are easy to read and supplying conversion algorithms to allow calculations of intervals. Since units of time smaller than "day" are often required, the general purpose representation for dates in the USL system is of the form 14.3.1975 12:05:34h (day month year hour minute second, with further provisions for fractions of seconds).

The precision necessary for the recording of dates depends on the subject matter: in nuclear physics it is different than in history or geology. The matter is further complicated by the fact that in the past the calendar has been changed several times (sometimes only regionally). This situation makes it practically impossible to devise an algorithm that calculates time intervals accurately for dates earlier than 1900. As a consequence, the precision given in the general purpose representation above may at times be unnecessary or even meaningless, e.g.,

ABC's dividends for 1975 were $7 per share. (6)

Here the date referred to is "1975"—"10/30/1975" would be incorrect. It might be argued that in (6) "1975" is not really a date but an interval, but this is also true for "10/30/1975," which stands for an interval from 0 am to 12 pm. Thus, dates may be considered as names for specific time intervals.

The answer to a question such as

Was Goethe a contemporary of Schiller? (7)

can be given without knowing month, day, or hour of either man's birth or death. In order to allow representation of different degrees of precision, all units except the year may be left out in the specification of a date in the USL

system, but if a smaller unit is specified, all greater units will be implied (see below) unless they are also provided. The time handling algorithms of the USL system take the precision given in a question to determine the precision required for the search in the data base: All units smaller than the ones specified are masked, but if a date in the data base is less precise than that specified in the question, it will not be put out as an answer.

A similar problem arises with respect to the representation of time intervals. For the display of intervals, there are basically two possibilities and both are implemented in the system. Intervals are displayed from greatest to smallest unit (5y03m15d 06:02:50h); or they can be displayed in a prespecified unit (63 [month]). In the latter case, there is a problem as to whether values should be rounded or truncated (as in the case of people's ages). The system currently provides only truncation.

The representation of repeated (e.g., periodic) events poses the greatest problem, and the solution currently implemented in the USL system can only be regarded as provisional:

John gets his wages every Friday. (8)

"every Friday" is stored *as is*. Although it is sufficient to answer the question

When does John get his wages?, (9)

some other mechanism would be needed to answer the question

Does John get his wages more often than Bill? (10)

A discussion is now given of how time is referenced in natural language and what kinds of operations must be performed to achieve usable representations. Consider the following examples:

John arrived in Brussels on Sept. 5, 1976, at 15:35. (11)

On Sept. 5, 1976, John arrived in Brussels at 15:35. (12)

In June Paul went to Rome. (13)

Bill will visit London in February. (14)

Mary met Jane today. (15)

Martin will arrive in three hours. (16)

Who arrived in Brussels before October? (17)

On what day of the week did John arrive in Brussels? (18)

Sentence (11) is certainly the easiest to handle; (12) poses some small problem because it has to be determined that date and time of day belong together. The system assumes that in (13) the last June before today is referred to, whereas in (14) the next February after today is taken; in both cases the appropriate year is added (this may not always be true in everyday conversations, but was considered realistic for interactions with a data base). The criteria employed are the occurrence in the temporal adverbial of a month but no year, and the tense of the verb. In addition, the month mentioned in the input sentence is compared with the current month in order to decide whether the previous, current, or following year should be used. Since German allows reference to future events with the present tense, no difference between present and future is made. In (15) "today" must be converted to an absolute date, since otherwise correct reference to the event would no longer be possible at a later date. Example (16) is similar to the previous three examples, with the addition that USL completes the date and calculates the time of day. Question (17) involves addition of the year and comparison of dates with the proper precision, whereas (18) requires conversion of the date to the appropriate day of the week.

The following examples refer to intervals:

Who arrived in Brussels between May and December? (19)

The meeting will last from 12:00 till 13:35. (20)

How long will the meeting last? (21)

How many minutes will the meeting last? (22)

Question (19) is treated like (17) except that two dates must be compared. When either (21) or (22) is put to the system after (20), the length of the interval is computed and displayed in the requested form.

A number of problems still remain that have not been addressed, because they were considered less urgent than the ones handled by the system. Problems to be addressed in the future are comparison of intervals with respect to relations like "contained in" and "overlap," which is done in [18]; proper treatment of tense and aspect instead of simple time references to past, present, and future; treatment of temporal clauses; and proper treatment of habitual, repetitive, periodic events.

Coordinated phrases

Coordinated phrases are phrases bound together by conjunctions such as "and," "or," and "neither . . . nor."

Which companies produce computers and typewriters? (23)

Which companies produce and sell computers? (24)

Winograd states in [28, p. 149] that one of the most complex parts of English is the system of conjunction. For German this statement holds equally true. Coordination is difficult because of both syntax and interpretation. Syntax analysis is difficult because those constituents of the sentence belonging to all coordinated elements usually

occur only once. Thus, those constituents of the sentence that are shared by the coordinated elements must be sorted out from those that are specific to just one of them. The interpretation is difficult, because the meaning of a conjunction depends very much on the context in which it stands. For example, if (23) were altered to

Which companies produce computers, and which companies produce typewriters? (25)

the meaning would be different, since in (23) companies are sought that produce *both* computers and typewriters, whereas in (25) companies are requested that produce *either* computers or typewriters. More precisely, two lists are expected, one containing those companies producing computers, and the other containing the companies producing typewriters. Companies satisfying (23) should appear in both lists. [Some speakers of English may feel that (25) is one reading of (23), but this is not important for the moment, since only the possible differences in meaning are explained.] In

List age, salary, and manager of all employees in department 405. (26)

the expected answer is a four-column table containing in each row an employee's identification, his age, salary, and manager. Thus "and" has a different meaning than in (23) and (25). In decision questions there are two cases possible:

Do some companies produce oil and electricity? (27)

Does Exxon produce oil, and does GM produce cars? (28)

Question (27) is to be interpreted like (23), but in (28) "and" may be considered to have still another meaning, i.e., *conjunction*, in the sense of propositional logic.

The difficulty in the interpretation process, therefore, consists in finding the proper criteria with which to interpret the different meanings of the conjunctions. To simplify the discussion, the following names will be introduced for the meanings of "and": a) Intersection [examples (23) and (27)], b) Multiple List [example (25)], c) Combined List [example (26)], and d) Conjunction [example (28)].

Note that additional meanings for "and" exist for sentence types other than questions. Unfortunately there is little discussion to be found on such meanings and the conditions for their occurrence, even though coordination is extensively dealt with in the literature (see, e.g., [35, pp. 294–418], [36]). A description of the interpretation of coordination in question-answering systems that are able to handle it [3, p. 202], [15, p. 335], [28, pp. 149ff., pp. 347ff.], [37, p. 903] was not available to the author.

The criteria that have been taken into consideration for the implementation of coordination in the USL system are the following:

1. type of sentence
   - complement question (wh-question) (23)
   - decision question (yes-no question) (27)
   - command ("list," "find," "show," etc.) (26)
2. type of coordination
   - sentence (25)
   - verb phrase (24)
   - noun phrase (23)
   - noun (26)
3. type of noun phrase
   - proper name ("Exxon")
   - interrogative pronoun ("who")
   - common noun phrase ("the companies")
4. position in clause
   - subject position [in (24), "which companies"]
   - nonsubject position [in (24), "computers"]
5. level of constituent
6. number
7. type of main verb
   - full verb
   - "to have"
   - "to be"

These criteria are not sufficient to unambiguously interpret all cases, since lexical meaning and world knowledge can come into play, as shown in [38]. In the USL system, when there is an ambiguity between Intersection and Multiple List or Conjunction, Intersection is taken, since there will always be alternate formulations possible for obtaining the other interpretations.

Criteria 1 and 2 are covered by the following general rules, which can easily be verified:

- In complement questions and commands all interpretations except Conjunction are possible.
- In decision questions, Intersection and Conjunction are possible interpretations.
- When complete sentences are joined by "and," only Multiple List (complement question) and Conjunction (decision question) are possible.
- When verb phrases are coordinated, the only interpretation possible is Intersection.

The effects of criteria 3–7 will only be illustrated by examples (the interpretation chosen by the system is given in parentheses).

Who is the manager of Brown and Jones? (Intersection) (29)

Who are the managers of Brown and Jones? (Multiple List) (30)

Questions (29) and (30) demonstrate the effects of criteria 3, 6, and 7, since "who" is singular, and only verbs like "be" and "become" can occur in plural forms with "who" as subject.

Where do Brown and Jones live? (Multiple List) (31)

Who lives in Berlin and London? (Intersection) (32)

The different interpretations of examples (31) and (32) are due to coordination in subject and object positions.

The effect of differences in the level of coordinated constituents can be seen in

List the employees who are male and unmarried. (33)

List the male and unmarried employees. (34)

In (33) "and" will be most likely interpreted as Intersection, whereas in (34) it tends much more to be interpreted as Multiple List, and these are also the interpretations chosen by the system.

The remainder of this section will describe the process of interpreting coordinated phrases in the USL system. During syntax analysis, coordinated phrases are handled in a relatively simple fashion by using rules such as

⟨NP [NCOORD(1,3,2)]⟩ ← ⟨NP⟩ ⟨CONJ⟩ ⟨NP⟩;

(Checks and assignments of syntactic features have been left out for the sake of simplicity.) The routine NCOORD builds part of an intermediate tree structure, as shown in example (5).

When this intermediate tree has been completed, it is scanned by a routine that produces as many trees as there are coordinated elements. In case of multiple coordination, the number of trees generated is the product of the numbers of elements in the respective coordinated phrases. A data base language expression is generated for each tree, and at the same time information is processed to determine the required interpretation. In the case of Intersection, the intersection is formed on the generated data base language expressions; for Multiple List, the respective lists are displayed in sequence; for Combined List, a routine is called that produces the combined table; and for Conjunction, the necessary number of *yes* and *no* answers is displayed. The latter is done because if part of the coordinate phrase is negated, it may become unclear whether *yes* or *no* should be the answer to the whole question.

Quantifiers

Quantifiers treated in the USL system include "the," "a," "all," "every," "each," "some," "no," "which," "how many" (and their German counterparts), and numbers. Since quantifiers interact with negation, it will be necessary to discuss both together. Quantifiers and negation are a source of confusion to native-language speakers, but they are generally not aware of the problems.

In other question-answering systems quantifiers are generally treated with heavy restrictions (see [13] for a comparison). The treatment in PHLIQA1 [39, 40] seems to be similar to the one presented here; however, the published details are not sufficient for an exact comparison to be made.

Two terms are defined that are not always consistently used in the literature. A *negated sentence* is a sentence containing "not" or some other particle of negation ("no," "never," etc.) (see, e.g., [36, p. 374]). The *negation* of a given sentence is a sentence having the opposite truth value.

The negation of

Besucht Jones Miller?
visits Jones Miller (35)

can be any of the negated sentences (36–38)

Besucht Jones Miller nicht? (36)

Besucht nicht Jones Miller? (37)

Besucht Jones nicht Miller? (38)

It does not make any difference to the truth value of a negated sentence where the "not" is put, if (and only if) there are only proper names in the sentence (differences in presuppositions are not considered here, as for example in [41]). The negation of

Besuchen alle Verkaeufer Miller?
visit all salesmen Miller (39)

is

Besuchen nicht alle Verkaeufer Miller? (40)

and

Besuchen alle Verkaeufer Miller nicht? (41)

Besuchen alle Verkaeufer nicht Miller? (42)

Questions (41) and (42) both mean

Does no salesman visit Miller? (43)

If "Miller" is replaced with "alle Kunden" ("all customers") in (39–42), (41) and (42) then mean in English respectively

Does no salesman visit a customer? (44)

Does no salesman visit all customers? (45)

The different translations of each series of the German sentences depend only on different word order, whose effects are differences in the *scopes* of the quantifiers and negation particles involved.

For natural languages, a definition of scope of quantifiers is difficult for two reasons: there exist no explicit variables in natural language sentences, and the quantifiers are not placed in a well defined position relative to their scope, but usually occur before nouns, with which they agree in number, gender, and case. As a con-

sequence, scope must be established by comparing the meanings of minimally differing sentences. Such a procedure is followed for the scope of negation particles by Hajicova [41, p. 138ff.].

The scope of quantifiers is defined stepwise, starting with very simple sentences. A sentence containing only one finite verb with its complements, no subordinate clauses, and no noun complements is called a *kernel sentence*.

Q1: Let k be a kernel sentence containing just one quantifier. Then the scope of the quantifier is k.

For example:

Does Jones control all orders? (46)

says something about all orders and their relationship to Jones. Thus "Jones" as well as "control" is within the scope of "all."

Q2: Let l be a kernel sentence with two quantifiers q and r. Then three cases are possible:
- a. The scope of q is wider than the scope of r; i.e., the scope of q is l, and the scope of r is l except for q.
- b. The scope of r is wider than the scope of q; i.e., the scope of r is l, and the scope of q is l except for r.
- c. The scopes of q and r are identical, namely l.

If q and r are the same quantifiers and are not numerical, then the distinction made is not important, since $AxAy\ P(x, y) \leftrightarrow AyAx\ P(x, y)$ and $ExEy\ P(x, y) \leftrightarrow EyEx\ P(x, y)$. If q and r are universal and existential quantifiers only (represented as A and E respectively), then case c is not important, because it amounts to $ExAy\ P(x, y) \wedge AyEx\ P(x, y)$, but due to $ExAy\ P(x, y) \rightarrow AyEx\ P(x, y)$, case c then corresponds to either case a or case b. The distinction made is relevant, however, for numerical quantifiers; consider

Do three managers need three secretaries? (47)

which could be synonymous with either of the following sentences:

Do exactly three managers need in total exactly three secretaries? (48)

Do exactly three managers need exactly three secretaries each? (49)

Question (48) corresponds to case c, and (49) shows that even if two numerical quantifiers are identical, they cannot in general be exchanged.

The problem is to ascertain what are precisely the conditions that determine the relative scopes of two quantifiers in a natural language sentence, taking also into account the scopes of negation particles that may be present. One possible hypothesis is [42, p. 5]:

Q3: The order of the scopes of quantifiers and negation particles in a kernel sentence is from left to right.

This hypothesis covers the vast majority of cases, both in English and German. It was therefore implemented as a rule in the USL system. However, there are many exceptions. They may be indicated by intonation [43, p. 458], by world knowledge counteracting the usual interpretation, or by special variants of quantifiers that suggest an unusual order of scopes, like "the same" (German: "derselbe"):

Do all employees work with the same manager? (50)

But

Do all employees work with the same manager as last year? (51)

has the usual order. However, neither intonation nor world knowledge are available in the USL system, and the behavior of special variants of quantifiers has not as yet been sufficiently investigated.

Kernel sentences with more than two quantifiers show basically the same problems, but they tend to be almost unintelligible, especially if they also contain a negation particle. They are interpreted according to Q3.

Sentences with subordinate structures such as relative clauses or other noun complements are of special interest, and the following rules were implemented:

Q4: The scopes of quantifiers in superordinate structures include subordinate structures.

Q5: The scopes of quantifiers in relative clauses are confined to the relative clauses.

Q6: The scopes of quantifiers in noun complements that are not clauses may include the clause to which their head nouns belong.

This last rule deserves an example:

Is there a manager of all employees? (52)

Are there managers for all employees? (53)

In (52) the order of quantifiers is as usual, but in (53) the order is reversed.

A further aspect is brought in by the quantifier "which." This quantifier may be interpreted as the lambda operator of predicate calculus (represented as Sx). If sentences (52) and (53) are changed to

Who is the manager of all employees? (54)

Who are the managers of all employees? (55)

then (54) may be characterized as $SxAy\ P(x, y)$, and then (55) can be represented by $SxEy\ P(x, y)$. Thus the "all" of

(55) has in fact become "some." This becomes clear when one considers the paraphrase

What is the set of managers who have at least one employee? (56)

which only a logician would use.

The interpretation of quantified phrases is done in the USL system by means of a special subroutine, since the data base language used is not powerful enough for expressing such phrases. The concepts underlying the design of this routine are described by Ott in [44].

Possessive pronouns

It is necessary to treat possessive pronouns in a question-answering system, since otherwise there exist questions to a given data base that cannot be formulated. For example,

Who earns more than his manager? (57)

cannot be paraphrased without using pronouns (e.g., "he") or demonstratives (e.g., "this") instead.

The difficulty in treating possessive pronouns (or pronouns in general) is to determine their reference. Although there are several publications about the reference of pronouns in generative transformational grammars (e.g., [35], [45–48], there are not many concerned with determining reference in the analysis of language ([28], [49–51]), despite its importance. Therefore, an empirical investigation was made of the rules that govern the reference of possessive pronouns.

First and second person pronouns were not considered, because they were not felt to be essential in our context. Syntactic structures not handled by our system (adverbial clauses, that-clauses, subject clauses, object clauses, and infinitive clauses) were also not considered. The remaining structures are

simple sentence	(who sells his car?)
coordinate noun phrase	(the officer and his wife)
coordinate verb phrase	(dated Jill and married her sister)
genitive attribute	(the wife of his manager)
prepositional phrase	(with his manager)
adverbial	(in their branch office)
quantified noun phrase	(all their employees)
relative clause	(manager who fires his clerk)

In order to simplify the discussion, it is useful to point out the possible types of reference:

1. Backward (anaphoric) reference (John sold his car.)
2. Forward (cataphoric) reference (Among his friends John is the most intelligent.)
3. Reference to previous sentence (His car was new.)
4. Reference to noun phrase within the same sentence (John sold his car.)
5. Reference to coordinate noun phrase (Jack and Jill like their manager.)
6. Reference to noun phrase in a higher clause (John dated the girl who married his brother.)

Reference to lower clauses is generally not possible. In

John dated the girl, who liked Mary, and married her sister. (58)

"her" cannot be construed to refer to "Mary" (see, however, the discussion of "dessen" below).

Reference to previous sentences was not considered, because no attempt has been made to implement the necessary devices into the present system. Forward reference is quite rare in both English and German (especially in questions). All other types of reference are quite common and have been taken into account.

There are two possessive pronouns in German with no direct correspondents in English, namely "dessen" and "deren." They can only refer to noun phrases in a nonsubject position, and it is even possible to use them to refer to lower clauses:

John schlug Bill und dessen Frau.
John beat Bill and his wife (Bill's). (59)

John zeigte dem Manager, der einen klugen Sohn
John showed to the manager who a bright son
hatte, dessen Preise.
had his awards (the son's). (60)

Dem Manager stahl John dessen Auto.
from the manager stole John his car (the manager's). (61)

"Dessen" and "deren" can often be translated as "of the former" or "of the latter," depending on the relative positions of subject noun phrase and nonsubject noun phrase. These pronouns are not handled properly as yet by the USL system, but they should be available in future versions.

The most fundamental (and trivial) rule about the reference of possessive pronouns is (see, e.g., [36, p. 369], [52, p. 616])

R1: Possessive pronouns and the noun phrases they refer to must agree in gender and number.

Those noun phrases in a sentence (except for the one containing the possessive pronoun) that agree in gender and number with the possessive pronoun are called *candidates for reference*.

If the reference is to a coordinate noun phrase and the conjunction is "and," agreement in number is to be interpreted such that the coordinate noun phrase is treated as

plural even if all its constituents are singular (see [36, p. 369, their example 39]).

Interesting and problematic are those cases of reference where an ambiguity exists. For the empirical investigation, syntactic structures were selected that allowed for more than one candidate for reference, such as

a. NP V NP and NP
John beat the officer and his wife.
b. NP V PP PP
John argued with Bill about his new project.
c. NP V NP[DAT] NP[ACC]
Hans stiehlt dem Manager sein Auto.
Hans steals from the manager his car.

For coordinate noun phrases and verb complements the following rule was found:

R2: Forward reference does not occur within coordinate noun phrases such that a possessive pronoun in an earlier constituent noun phrase refers to a later one. The same is true for the reference among verb complements.

A similar rule is given in [36, p. 554].

It is not surprising to note that not all sentences containing more than one candidate for reference for a possessive pronoun are actually ambiguous. This is so for semantic or pragmatic reasons. Consider

John married the manager to his daughter. (62)

If (62) refers to an event in a western society and no special circumstances have been explicitly mentioned before, "his" must refer to "John" rather than "manager," since people are not allowed to marry their own daughters. This reasoning is best characterized as *pragmatic*. Consider

John asked the manager for his permission. (63)

John verweigerte dem Manager seine Erlaubnis.
John refused to the manager his permission. (64)

In (63) "his" clearly refers to "manager." Sentence (63) is different from (62) in that no overriding context is found for (63) that makes reference to "John" possible. Sentences (63) and (64) together show that the referent of "his" is determined by the meanings of the pairs ("ask," "permission") and ("refuse," "permission"), rather than by the meaning of "permission" alone. Such an argument is called *semantic*.

Different interpretations of the referent of 'his' would make (63, 64) semantically and (62) pragmatically anomalous. This situation can be described by a compatibility rule already mentioned in [51] as SRR1 in a slightly different form:

R3: A possessive pronoun refers to a candidate for reference only if no semantic or pragmatic anomaly is thereby generated.

If sentences are looked at in isolation, as in the USL system, the semantic or pragmatic information required often cannot be obtained. In addition, not all information that could theoretically be used can be properly encoded to permit the reasoning required to resolve the reference. In an interactive system, the user can be involved in the process of reference resolution, but this means that the analysis is no longer completely formal. Such a procedure is justified, however, because in human dialogue the hearer also asks back, when he cannot resolve the reference himself.

In the USL system, a mechanism was implemented that observes the syntactic rules R1 and R2. When it detects that more than one candidate for reference remains, the system displays them as well as the head noun of the possessive pronoun; the user then types in the number of the referent he had in mind. Since no semantic or pragmatic criteria are used, sentences such as (62)–(64) would be treated as ambiguous. In Winograd's system, requests for clarification are also put to the user ([28, p. 371ff.]), but, in addition to syntactic criteria, a plausibility mechanism is used for the determination of the reference of pronouns.

Once the proper referent has been found, the data base language statement is generated accordingly. Here it is necessary to remember that nouns generally refer to relations, and that these relations often have a so-called VON-column (see section on semantic concepts). A sentence such as

Is John selling his car? (65)

can be paraphrased as

Is John selling John's car? (66)

Under the assumption that *sell* and *car* are relations having the following shapes:

SELL(NOM,ACC,TO), CAR(NOM,VON),

(66) can be translated as

((SELL;C1='JOHN')*(CAR;C2='JOHN');C2=C4) (67)

This translation is unfortunate because the sentence

Who is selling his car? (68)

cannot be translated in the same way. However, translation (69) is also possible, and it can easily be modified to represent (68):

((SELL;C1='JOHN')*CAR;C2=C4);C1=C5) (69)

((SELL*CAR;C2=C4);C1=C5) (70)

Similar solutions are possible in all cases where no coordinate noun phrases occur. As mentioned previously, coordinated phrases are separated out by a sort of copying transformation, and separate data base language expressions are generated for them. Since reference is possible between the respective trees, an additional operation is necessary after the simple data base language expressions have been formed.

Quantified noun phrases like "all their employees" cannot as yet be interpreted, and they pose a number of new problems if they occur in coordinated phrases. Since structures of this kind are quite common, solutions must be found for them in future versions of the system.

Conclusion

The USL system is an experimental system whose evaluation with real applications and users has begun. Users' response to the system is being analyzed (e.g., whether they are willing to live with its limitations and in what directions improvements should be made).

The linguistic problems addressed have all been investigated before by others, and in no case has a complete solution been found. However, it is hoped that the work on the USL system has provided some new insights into the problems.

The discussion of temporal references concentrated on the conversion of deictic references to actual dates, and on the treatment of vagueness of time references. Both features improve the *user friendliness* of the system.

As a result of work on the coordination problem, four different meanings of "and" were isolated and the syntactic criteria influencing their selection given. These meanings are also considered relevant for the interpretation of questions outside the context of access to data bases.

The notion of scope of quantifiers was defined for natural language sentences, and the interaction between quantifiers and negation particles was demonstrated.

The mechanisms governing the reference of possessive pronouns were outlined. Not all the information necessary to determine such references is available in the USL system; hence, in some cases, requests for clarification must be put to the user.

Experience with the USL system suggests that such a system can also be regarded as a valuable tool for testing linguistic hypotheses and gaining knowledge about language which otherwise would be very hard to verify.

Acknowledgment

This paper describes work done on the USL project by my colleagues N. Ott, M. Zoeppritz, and myself.

References

1. E. F. Codd, "Seven Steps to RENDEZVOUS with the Casual User," *IBM Research Report RJ 1333*, IBM San Jose Research Laboratory, 1974; also, *Data Base Management*, J. W. Klimbie and K. L. Koffeman, eds., North-Holland Publishing Co, Amsterdam, 1974, p. 179.
2. J. Mylopoulos, A. Borgida, P. Cohen, N. Roussopoulos, J. Tsotsos, and H. Wong, "TORUS—A Natural Language Understanding System for Data Management," *Proceedings of the 4th International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1975.
3. E. D. Sacerdoti, "Language Access to Distributed Data with Error Recovery," *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977.
4. D. L. Waltz and B. A. Goodman, "Writing a Natural Language Data Base System," *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977.
5. M. Sibuya, T. Fujisaki, and Y. Takao, "Noun-Phrase Model and Natural Query Language," *IBM J. Res. Develop.* **22,** 533 (1978, this issue).
6. M. Kay, "Experiments with a Powerful Parser," *Proceedings of the Second International Conference on Computational Linguistics*, Grenoble, August 1967.
7. B. H. Dostert and F. B. Thompson, "The Syntax of REL English," *REL Report No. 1*, California Institute of Technology, Pasadena, 1971.
8. B. H. Dostert and F. B. Thompson, *Verbal Semantics in a Relational Data Base System*, California Institute of Technology, Pasadena, 1973.
9. F. B. Thompson, P. C. Lockemann, B. H. Dostert, and R. S. Deverill, "REL: A Rapidly Extensible Language System," *Proceedings of the 24th National Conference of the ACM*, New York, 1969, p. 399.
10. R. F. Simmons, "Natural Language Question-Answering Systems: 1969," *Commun. ACM* **13,** 15 (1970).
11. K. Sparck Jones and M. Kay, *Linguistics and Information Science*, Academic Press, Inc., New York, 1973.
12. D. E. Walker, "Automated Language Processing," *Annual Review of Information Science and Technology*, Vol. 8, C. A. Cuadra, ed., AFIPS, Washington, 1973.
13. R. Kogon, D. Lattermann, H. Lehmann, N. Ott, and M. Zoeppritz, "The User Specialty Languages System," GI-6. *Jahrestagung (Proceedings of the 6th Annual Meeting of the Gesellschaft fuer Informatik)*, Berlin, 1976, p. 221.
14. W. J. Plath, "Transformational Grammar and Transformational Parsing in the REQUEST System," *Proceedings of the International Conference on Computational Linguistics*, Pisa 27 VIII-1/IX, 1973, Vol. I, Casa Editrice Olschki, Firenze, Italy.
15. W. J. Plath, "REQUEST: A Natural Language Question-Answering System," *IBM J. Res. Develop.* **20,** 326 (1976).
16. W. A. Woods, R. M. Kaplan, and B. Nash-Webber, "The Lunar Sciences Natural Language Information System: Final Report," *BBN Report 2378*, Bolt, Beranek & Newman, Inc., Cambridge, MA 1972.
17. S. R. Petrick, "On Natural Language Based Computer Systems," *IBM J. Res. Develop.* **20,** 314 (1976).
18. B. C. Bruce, "A Model for Temporal References and Its Application in a Question Answering Program," *Artificial Intelligence* **3,** 1 (1972).
19. S. J. P. Todd, "The Peterlee Relational Test Vehicle—A System Overview," *IBM Syst. J.* **15,** 285 (1976).
20. O. Bertrand, J. J. Daudennarde, D. Starynkevitch, and A. Stenbock-Sermor, "User Application Generator," *Proceedings of the IBM International Technical Conference on Relational Data Base Systems*, Bari, Italy, 1976, p. 83.
21. H. Lehmann and N. Ott, "Interpretation Routines for German Grammar Rules," *Technical Note TN 75.03*, IBM Heidelberg Scientific Center, Heidelberg, Germany, 1975.
22. H. Lehmann, N. Ott, and M. Zoeppritz, "Language Facilities of USL/German," Version III, *Technical Note TN 77.04*, IBM Heidelberg Scientific Center, Heidelberg, Germany, 1977.

23. H. Lehmann and M. Zoeppritz, "Grammar Rules for German," Version II, *Technical Note TN 75.02*, IBM Heidelberg Scientific Center, Heidelberg, Germany, 1975.
24. H. Lehmann and M. Zoeppritz, "Partition of German Grammar," *Technical Note TN 75.05*, IBM Heidelberg Scientific Center, Heidelberg, Germany, 1975.
25. H. Lehmann and M. Zoeppritz, "Grammar Rules with Examples," *Technical Note TN 75.06*, IBM Heidelberg Scientific Center, Heidelberg, Germany, 1975.
26. D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd, "GUS, A Frame-Driven Dialog System," *Artificial Intelligence* **8,** 155 (1977).
27. E. Charniak, "Inference and Knowledge II," *Computational Semantics*, E. Charniak and Y. Wilks, eds., North-Holland Publishing Co., Amsterdam, 1976, p. 129.
28. T. Winograd, "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," *Project Mac TR-84*, Massachusetts Institute of Technology, Cambridge, MA, 1971.
29. Y. Wilks, "Parsing English I," *Computational Semantics*, E. Charniak and Y. Wilks, eds., North-Holland Publishing Co., Amsterdam, 1976, p. 89.
30. R. Carnap, *Meaning and Necessity*, Chicago University Press, Chicago, IL, 1947.
31. H. Lehmann, *Linguistische Modellbildung und Methodologie (Linguistic Methodology and Model Building)*, Niemeyer, Tuebingen, Germany, 1973.
32. H. Lehmann, "The USL System—Its Objectives and Status," *Proceedings of the IBM International Technical Conference on Relational Data Base Systems*, Bari, Italy, 1976, p. 7.
33. R. Montague, "The Proper Treatment of Quantification in Ordinary English," K. J. J. Hintikka, J. M. E. Moravcsik, and P. Suppes, eds., *Approaches to Natural Language*, Reidel Publishing Co., Dordrecht, Netherlands, 1973, p. 221.
34. D. Lewis, "General Semantics," *Semantics of Natural Language*, D. Davidson and G. Harman, eds.: Reidel Publishing Co., Dordrecht, Netherlands, 1972, p. 169.
35. R. P. Stockwell, P. Schachter, and B. H. Partee, *The Major Syntactic Structures of English*, Holt, Reinhart and Winston, Inc., New York, 1973.
36. R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik, *A Grammar of Contemporary English*, Longman, London, 1972.
37. L. R. Harris, "ROBOT: a High Performance Natural Language Data Base Query System," *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977.
38. N. Ott, "Problems in the Interpretation Process of the USL System," *Technical Note TN 76.03*, IBM Heidelberg Scientific Center, Heidelberg, Germany, 1976.
39. S. J. P. Landsbergen, "Syntax and Formal Semantics of English in PHLIQA1," *Philips Report M.S. 9572*, Eindhoven, Netherlands, 1976.
40. R. J. H. Scha, "Semantic Types in PHLIQA1," *Philips Report M.S. 9577*, Eindhoven, Netherlands, 1976.
41. E. Hajicova, *Negace a presuposice ve vyznamove stavbe vety (Negation and Presupposition in the Semantic Structure of the Sentence)*, Charles University Press, Prague, 1975.
42. P. B. Sheridan, "On Dealing with Quantification in Natural Language Utterances," *Research Report RC 6422*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1977.
43. G. Helbig and J. Buscha, *Deutsche Grammatik, Ein Handbuch fuer den Auslaenderunterricht (German Grammar, a Handbook for Teaching Foreigners)*, VEB Verlag Enzyklopaedie, Leipzig, GDR, 1974.
44. N. Ott, "Interpretation of Questions with Quantifiers and Negation in the USL System," *Technical Report TR 77.10.005*, IBM Heidelberg Scientific Center, Heidelberg, Germany, 1977.
45. L. Karttunen, *Discourse Referents*, Indiana Linguistics Club Publications, Bloomington, 1975.
46. G. Lakoff, *Pronouns and Reference*, Indiana Linguistics Club Publications, Bloomington, 1968.
47. G. Lakoff, *Counterparts, or the Problem of Reference in a Transformational Grammar*, Indiana Linguistics Club Publications, Bloomington, 1968.
48. B. H. Partee, "Opacity, Coreference, and Pronouns," *Semantics of Natural Language*, D. Davidson and G. Harman, eds., Reidel Publishing Co., Dordrecht, Netherlands, 1972, p. 415.
49. I. Batori, R. Henning, H. Lehmann, B. Schirmer, and M. Zoeppritz, "LIANA—Ein deutschsprachiges Frage-Antwort-System," ("LIANA—A German Language Question Answering System), *ITL (Institut voor Toegepaste Linguistiek) Rev. Appl. Linguistics* **30,** 1 (1975).
50. K. Ebert, "Zur automatischen Erkennung von Referenzidentitaeten" ("On Automatic Recognition of Referential Identity"), paper read at the 5th annual meeting of the Gesellschaft fuer angewandte Linguistik, Stuttgart, Germany, 1973.
51. H. Lehmann, "Kontextuelle Referenz" ("Contextual Reference"), paper read at the 6th annual meeting of the Gesellschaft fuer Angewandte Linguistik, Stuttgart, Germany, 1974 (unpublished).
52. P. Grebe, H. Gipper, M. Mangold, W. Mentrup, and C. Winkler, *DUDEN—Grammatik der deutschen Gegenwartssprache (DUDEN—Grammar of the Contemporary German Language)*, Mannheim, Germany, 1973.

Received September 1, 1977; revised May 9, 1978

The author is located at the IBM Germany Scientific Center, Tiergartenstrasse 15, 6900 Heidelberg, Germany.