# The Meaning of OF and HAVE in the USL System

Magdalena Zoeppritz

IBM Germany
Heidelberg Scientific Center
Tiergartenstrasse 15
D-6900 Heidelberg WEST GERMANY

**This paper shows how the transformational relationship between HAVE-sentences and OF-phrases is used to represent data contained in sentences with HAVE as the main verb in the context of an information system using natural language to access a relational data base. An overview of the system first establishes the framework in which natural language processing is done. Then ways of representing HAVE are discussed with emphasis on the relation between HAVE and OF. The interpretation proposed and the interpretation process are illustrated by a list of representative queries and phrases against a small data base. In conclusion, this interpretation is extended to prepositional attributes with WITH and WITHOUT, and problems are discussed.**

## 1. Introduction

The User Specialty Languages (USL) System translates input in natural language German into expressions in the formal language of the data base system associated with USL, accesses the data base with these expressions and transmits the results to the user either directly or after performing additional operations on the output. The system was developed at the Heidelberg Scientific Center of IBM Germany by H. Lehmann, N. Ott, the students K. Horländer and W. Sauermann, and the author.

USL was designed to provide data base access for user groups whose requirements are not satisfied by standard programs and for whom having special programs written or learning to program themselves would not be feasible.

The system was to be capable of dealing with natural language in a variety of different application contexts and not restricted to a particular field or world. This purpose determined the methods used and the corresponding limitations of the system. Within these limits, we have tried in the implementation to incorporate the syntactic constructions to be expected in the context of data base interaction and to provide the correct interpretations for them.

The data base management system is an implementation of the relational model, the Peterlee Relational Test Vehicle (PRTV, Todd, 1975) with its data base language ISBL. This is the target language of the translation process. The method of translation is a substantial extension of the methods used in the REL System (Rapidly Extensible Language, Thompson et al., 1969, Dostert et al., 1971). This system treats natural language much like a formal language in that syntactic constructions and function words are interpreted by the system according to the semantics of the language built into the system, but nouns, verbs, and adjectives are treated as variables of which only the data type — the word class — is known.

The underlying assumption is the following: The meanings of prepositions, dates, verbs like HAVE and BE and syntactic constructions, on the one hand, are independent of the subject matter; on the other hand, nouns, verbs, and adjectives and their meanings vary from application to application. In the context of a given data base, these words identify the names of relations; their meaning is restricted to the association between word and corresponding relation. Names and numbers identify values within tuples.

In USL, these words can be added to the system by the user to match the shape of the relations in his data base. A prompting routine guides the vocabulary definition and makes sure that all information needed by the system is entered.

The advantage of not providing the user with the vocabulary of his application already built into the system is that he is not restricted in his choice of words, and new words can be added easily. The dis-

0362-613X/81/020109-11$01.00

advantage is that the system knows only what has been explicitly defined by the user, and common sense knowledge, e.g. that an employee is a person or that a salary does not own a house, is not reflected in the system and is not used in the interpretation process. This also means that the interpretation can make only very restricted use of deep case relationships, because the information on deep cases would have to be obtained from the user. We see no way as yet to elicit this information reliably and consistently, without confusing or boring the user.

When used for an application, then, the system works with two vocabularies, one user-defined, containing the nouns, verbs, and adjectives referring to the data base of his application, the other system-defined, containing prepositions, quantifiers, interrogatives, particles, names of days and months, nouns referring to operations like minimum, maximum, plus, minus, etc., the verbs BE and HAVE, and adjectives like GREATER, MORE, and LESS. The system-defined words and their meanings to the system are the same for all applications.

The purpose of this paper is to discuss and demonstrate the interpretation of one of these system-defined words: the verb HAVE when used as the main verb of a sentence. (HAVE as an auxiliary has not been implemented.) An overview of the system first establishes the framework in which this interpretation has its place. This is followed by a discussion of HAVE in general and possible representations of it in USL. The solution proposed here and implemented in USL uses the transformational relationship between HAVE-sentences and OF-phrases to represent and search data contained in HAVE-sentences in the relation addressed by the accusative of HAVE. This interpretation is applied to a list of representative queries and phrases against a small data base as a test of the interpretation and as an illustration of the interpretation process. In the concluding sections, the interpretation is extended to prepositional attributes with WITH and WITHOUT, and problems are discussed.

## 2. The System

This section contains a brief overview of the system. More detail, particularly on the interpretation, is found in Ott (1977, 1978), and Lehmann (1978). Information on language coverage is found in Lehmann et al. (1977), and documentation on the grammar is in preparation.

The system consists of three parts:

1. Parser and grammar rules
2. Function processor and interpretation routines
3. Data base and data base language

The system contains about 800 syntactic and 200 lexical (system-defined words) rules. The rules are mostly context-free, with some context-sensitive and some transformational rules, written in a modified Backus Normal Form (BNF). Each rule contains the name of an interpretation routine written in PL/1, which performs operations in correspondence to the meaning of the syntactic construction.

The parser and function processor are USAGE (User Application Generator), developed at the IBM Paris Scientific Center, along the principles of Kay (1967), with considerable modifications and extensions. It works from left to right, bottom-up, through the input string. The input is tested against the rules, and rules are applied wherever there is a match, lexical rules first. The result is several disconnected subtrees, which are discarded, and one or more trees spanning the entire input. Only the full parses are processed further. Each node in the tree contains the name of the interpretation routine from the grammar rule used in its construction.

The function processor walks down the tree and calls the interpretation routines associated with each node. In the original concept developed for REL, the interpretation routines were executed on the spot and the result was passed as an input parameter to the next routine. This procedure proved insufficient for the interpretation of quantifiers, negation, and coordinated structures. The original concept was changed so that now the interpretation routines do not simply pass on a result, but successively build a structure reflecting semantic dependencies indicated by the syntactic structures, the names of the relations, columns and values taking part in these dependencies, as well as information on syntactic function and scope of individual elements in the tree. This structure is processed recursively and translated into expressions in the data base language ISBL (cf. Ott, 1979, and Lehmann, 1978, for detail on the interpretation process).

The resulting ISBL strings are passed to the database management system to access the data base and perform the update or retrieval operations requested. In the simple case, if the answer is a list or table of items, and the question calls for nothing more, the answer is formatted and printed. For yes/no questions, the return code from the data base is translated to the proper answer. Questions involving some types of quantification and arithmetic need further processing on the answer.

The data base is relational. Relations can be thought of as tables with rows and columns. Noun, verbs, and adjectives refer to relation names; they can be words in the language or words invented for a specific purpose. Names of columns within relations, so-called role names are standardized in USL. They correspond to the complements of the nouns, verbs, or adjectives. This often coincides with their valence, but is not valence in the strict sense, because it does not matter in USL whether complements are obligatory or optional in the language, but only whether these complements are needed in the application. Thus, a relation SUPPLIER can be defined with two, three, or more columns, depending on the data and purposes of the data base containing it, e.g.:

supplier of product
supplier of product to recipient
supplier of product to recipient at time

The standard role names are:

| | |
|---|---|
| NOM | nominative with verbs, with nouns and adjectives, set of objects referred to |
| ACC | accusative |
| DAT | dative |
| GEN | genitive |
| OF | genitive attribute |
| LA | place |
| LO | origin |
| LG | goal |
| LD | distance |
| LP | path |
| TA | point in time, date |
| TO | start |
| TG | end |
| TD | duration |
| (preposition) | e.g. FUER, name of preposition governed |

Thus, a sentence like:

Joan is the daughter of Harry

makes the system look for a relation DAUGHTER with two columns, NOM and OF, to add the tuple Joan, Harry.

| DAUGHTER | |
|---|---|
| NOM | OF |
| . | . |
| . | . |
| Joan | Harry |

The words IS, THE, and OF, as well as the constructions SUBJECT-OF and PREDICATE-NOMINAL-OF, are understood by the system. DAUGHTER is known to be a noun and therefore a relation name. JOAN and HARRY are unknown strings and therefore assumed to be values in a relation. Within this framework, the verb HAVE is one of the system-defined words. The following section discusses the interpretation of HAVE in the USL system and the reasons for this interpretation.

### 3. The Interpretation of HAVE

Compared with the extensive discussion of the verb BE, which is accorded special treatment both in linguistics and logic, the verb HAVE does not seem to have appeared in any way problematic. Syntactic peculiarities have been observed — a transitive verb which does not readily admit the passive — and a wide range of meanings are given in any dictionary, many of them idiomatic. For sentence analysis within a data base context, the major question is that of where to store information contained in sentences with HAVE as the main verb and from where to retrieve it. In the interpretation found in Cresswell (1973) or Bennet's (1974) extension of Montague, HAVE appears as a two-place predicate.

This interpretation is also widely accepted in artificial intelligence. But, as we are going to show, the interpretation leads to incorrect results and should be abandoned.

HAVE as a two-place predicate in a relational data base would have to be set up as a relation HAVE with two columns, one for the subject, one for the objects of HAVE-sentences. The tuples of the relation would contain the individuals (names, part numbers, figures) among which the relation HAVE holds. However, a closer look at the contents of such a relation shows that two places, one for the subject and one for the object, are not enough. Given the facts that:

John has a secretary by the name of Pauline

John has a daughter named Polly, who is a secretary

The corresponding data base entries in the relations HAVE, DAUGHTER, and SECRETARY would read:

| HAVE | | DAUGHTER | | SECRETARY | |
|---|---|---|---|---|---|
| NOM | ACC | NOM | OF | NOM | OF |
| John | Polly | Polly | John | Polly | Bill |
| John | Pauline | | | Pauline | John |

From this would follow correctly that "John has (daughter) Polly" and "John has (secretary) Pauline" but also that "John has two secretaries", because both Polly and Pauline are secretaries. The relationship between elements expressed as "x has y" is too vague; it can apply in too many cases, and can often be reversed "if x has (daughter) y, then y has (parent) x"; so that at least the specific relationship that makes it possible to speak of x having y would have to be recorded in a third column: "x has y as z". In that case, the examples above would not lead to John's having two secretaries:

| HAVE | | |
|---|---|---|
| NOM | ACC | AS |
| John | Pauline | secretary |
| John | Polly | daughter |

What is written in the third column sometimes appears overtly in sentences, but it is not part of the valence of HAVE, so that it would be difficult to require HAVE to be used only with reference to the relationship in the third column. Furthermore, the fact that the items in the third column are names of relationships and not names of individuals is significant in itself. This leads to the conclusion that HAVE should not be regarded as a primitive predicate at all, but as a derived predicate, derivable just in case some other relationship exists between the individuals in question.

With different aims and from a different point of view, this has already been observed by Bach (1967:476-77):

> It has often been said that *be* has no meaning by itself but only in connection with *Predicate*, the passive construction, and so on. The same is true of *have*. The two forms are distinguished syntactically from most true verbs by the fact that they have no selectional restrictions in themselves, but occur in constructions where the selections reach across from "subject" to "object" or complement. Likewise, from a semantic point of view, their contribution tó the meaning of the sentence is determined completely by the items that they link.

Conversely, HAVE can only be used meaningfully to link elements where some other relationship determining the nature of that link is expressed or can be inferred. The vague term "some other relationship" needs more clarification than can be given at this time. It is clear that the relationship must be representable as a two-place predicate, but many such predicates do not serve as the basis for deriving HAVE. It seems, for instance, that the relations expressed by action verbs do not permit HAVE to be derived directly, though their agent nominalizations often do:

John teaches Jack
≠ John has Jack ≠ Jack has John
John is the teacher of Jack
= Jack has John as his teacher

The relationships most often associated with HAVE are those of possession and ownership. That the meaning of HAVE is much wider is commonplace. (A detailed analysis of the syntactic properties of HAVE and their association with different meanings is found in Pitha 1971 and 1972). Still, the verbs OWN and POSSESS can be replaced by HAVE without intervening nominalization and the extension of

John has a bicycle

from

John owns a bicycle

to

John has a bicycle in his possession

seems artificial. On the other hand, the conclusion:

If John has a bicycle, then he owns a bicycle

is plausible and often true, but not necessarily so, and will be false with different choices of objects. In the case of inalienable possession as the objects of HAVE sentences, the conclusion is absurd, so that the objects of HAVE and OWN are taken to refer to different entities (Bierwisch, 1965). Nevertheless, while it is clearly impossible to restrict the interpretation of HAVE to possession, any other interpretation will face the problem that HAVE is often used as a synonym of OWN or POSSESS (cf. section 6). This may be the reason why HAVE is often regarded as a primitive for possession (e.g. Langacker, 1975). There is a transformational relationship between HAVE and OF, already discussed e.g. in Bach (1967):

Peter has a daughter Joan
Joan is the daughter of Peter

But there is a peculiarity: Whereas HAVE can be used with names, as well as with names and common nouns, whereby a relationship is implied if if is not overtly expressed (e.g., We each have our own room, I have A101), OF is rarely used only with names:

Peter has A202
?(The) A202 of Peter

but

Peter has a room
The room of Peter

This seems to show that, unlike HAVE-sentences, OF-phrases are acceptable only where the relationship between individuals is not implied but overtly stated. There is no base relationship underlying the use of OF, the base relationship is the one preceding OF in the OF-phrase. Furthermore, OF-phrases seem to be the specific means to express such relationships. The "secretary of Peter" is the individual which is related to Peter via the relationship "secretary of". OF-phrases cannot be expanded in the same way as HAVE-sentences can be expanded by AS-complements to introduce the "real" base relationship. And, because the base relationship is overtly stated, OF-phrases cannot be reversed:

?(The) Peter of the secretary

In this sense, OF-phrases can be considered as being more explicit than the HAVE-sentences into which they can be transformed.

This led to the decision to interpret HAVE sentences in USL as transformations of OF-phrases and consequently to search or store information in HAVE-sentences not in a relation representing the verb, but in the columns with the role-name OF in relations addressed by the nouns in the sentence.

The following uses of OF are not transformationally related to HAVE and are excluded here:

Helen *of* Troy
piece *of* chalk
distance *of* 3 miles
love *of* God
angel *of* a nurse
the destruction *of* the city
basket (full/out) *of* wood
man *of* property

For the purposes of data base query, some of these constructions with OF do not seem to be necessary. Measure expressions are desirable, but have not been implemented. An unsolved problem is how OF-phrases resulting from nominalizations of verbs can be related within the USL framework in a general way to the verb or to the event referred to. The remaining uses of OF have a range of meaning similar to HAVE in expressing not only possession and part-of relationships, but also any number of other relationships which, in the case of OF, are explicitly named by the noun preceding OF and, in the case of HAVE, can be inferred from the nouns or occur explicitly in the AS-complement.

All uses where OF is related to HAVE, as well as some of the other uses, can in German also be expressed by genitive attributes (not all are possible in English because of the restricted use of the genitive). The uses related to HAVE also appear as possessive pronouns. Genitive attributes, as well as possessive pronouns, are interpreted in the same way as OF-phrases. An additional selection operation for possessive pronouns is necessary to obtain a match between the individual members of the sets referred to by the possessive and addressed by the head noun of the possessive.

If the transformational relationship between HAVE-sentences and OF-phrases is to be used, and HAVE-sentences access the OF-column of relations (the column with the role name OF), there are still two possibilities: Given the sentence:

Which manager has a secretary?

a first interpretation addresses the OF-column of MANAGER and compares the contents of that column with the list of secretaries: "Is there a manager of somebody, is that somebody a secretary, and if so, who is the manager":

The set of x, such that x is a manager of y
and y is a secretary:

$$\{x \mid Ey(M(x,y) \wedge S(y))\}$$

This interpretation is valid only where secretaries are managed by the people whose secretary they are, but fails for:

Which manager has room 35?

unless the room of the manager is also contained in the OF-column of MANAGER. In general, this interpretation will work only where everything a manager can have is contained in the OF-column of MANAGER and is therefore not useful as a general solution. Furthermore, the interpretation is unable to handle correctly sentences like:

Which manager has a musician as his secretary?
Which manager has a secretary as his musician? [1]

It assigns the following interpretation to both sentences:

$$\{x \mid Ey(M(x,y) \wedge S(y) \wedge Mu(y))\}$$

though their meaning is clearly different.

A second interpretation of the sentence below:

Which manager has a secretary?

searches in the OF-column of the accusative, SECRETARY, for an entry that is also listed in MANAGER: "Is there a secretary of somebody, is that somebody a manager, and if so, who is it":

The set of x such that there is a y who is
a secretary and y is secretary of x and x is
a manager:

$$\{x \mid Ey(S(y,x) \wedge M(x))\}$$

Accordingly:

Which manager has room 35?

The set of x such that the room of x is 35
and x is a manager:

$$\{x \mid A(x,35) \wedge M(x)\}$$

The second solution does not require that everything a manager can have is found in the OF-column of manager. The selection starts with the relation named by the direct object of HAVE. In this way it is also guaranteed that the relation specified actually obtains between the respective individuals. In the example:

Which manager has a secretary?

the secretary requested is not just any secretary, but the secretary of this manager, not perhaps a colleague of this manager and the secretary of another.

This interpretation also distinguishes between the two sentences with AS-complements, whereby the complement takes the place of the accusative and the accusative is treated as an apposition to it.

Which manager has a musician as his secretary?

The set of x such that y is secretary of x
and y is a musician and x is a manager:

$$\{x \mid Ey(S(y,x) \wedge M(x) \wedge Mu(y))\}$$

Which manager has a secretary as his musician?

The set of x such that y is musician of x
and y is a secretary and x is a manager:

$$\{x \mid Ey(Mu(y,x) \wedge M(x) \wedge S(y))\}$$

The first solution could be used as an escape if the other interpretation does not yield a result, but it leads to multiple interpretations with the same result for relations that are defined as converses of one another. In other cases, this interpretation leads to answers where the answer should be undefined:

Which manager has 5000?
(salary, personnel, or what)
Which manager has A202?
(room, car, personnel number)

As a result, the second interpretation has been implemented in its strict form: There is generally no answer defined for queries in which the object of HAVE does not contain a relation. In human dialogue, such questions can often be answered because either it is clear that HAVE means POSSESS or BE PART OF, or

[1] The possessive pronoun here and in the examples below makes for better reading of the English glosses, the German examples do not have it and it is not necessary for the discussion.

because what corresponds to the relation appears from the context.

## 4. Test of the Interpretation

In order to see whether relating the subject of HAVE to the OF-column of the accusative of HAVE is a correct and general solution to the problem of interpreting HAVE, this interpretation was tried out with sentences containing HAVE and six types of noun phrases relevant in USL: names, quantified common nouns, common nouns preceded by interrogatives, relative pronouns, interrogative pronouns, and noun phrases with apposition, all both in subject and object positions. Coordinated noun phrases were not tested separately, because they are expanded into as many separate sentences as there are noun phrases in the coordination. Similarly, where there is no negation, common nouns preceded by quantifier or preceded by interrogative do not require different interpretations with respect to HAVE. Quantifiers trigger several transformation operations on the ISBL-string resulting from the translation (Ott, 1977), but the translation is the same. Interrogatives indicate which columns of the result are to figure in the answer. Appositions of the type "secretary Moser", "Moser as secretary" (including AS-complements of HAVE) have been included, because the first type furnishes a selection from the relation addressed, and the second type the relation itself.

The relevant features then are: name, noun, interrogative, referent of relative pronoun, and negation. The following section lists the test phrases for these cases, but not all their permutations. Also, appositions are not shown for relative clauses, for the sake of brevity. The list illustrates how the interpretation of HAVE outlined here is implemented in USL. For each case, the general ISBL expression which results from the interpretation of HAVE is formed. Then examples are formulated against a sample data base and translated into ISBL expressions according to the general schema, and the results of the data base operations triggered by the ISBL expression are shown, together with the columns taking part in the selection operations. The examples have been left in German, with glosses in English, because some of the constructions tested cannot be formulated in English in the same way.

The sample data base contains the following relations:

| MANAGER | | SECRETARY | | MUSICIAN |
|---|---|---|---|---|
| NOM | OF | NOM | OF | NOM |
| Stern | Moser | Moser | Sauer | Moser |
| Stern | Mahle | Mahle | König | Pahle |
| Sauer | König | Maier | Küfer | Peter |

Notation:

| | |
|---|---|
| NOM | relation/value in the nominative of HAVE |
| ACC | relation/value in the accusative of HAVE |
| APP | relation/value in the apposition |
| NOMnom | NOM-column of the relation in the nominative |
| OFnom | OF-column of the relation in the nominative |
| NOMacc | NOM-column of the relation in the accusative |
| OFacc | OF-column of the relation in the accusative |
| NOMapp | NOM-column of the relation in the apposition |
| OFapp | OF-column of the relation in the apposition |
| NOMwh | NOM-column of the relation with interrogative |
| NOMrel | NOM-column of the relation governing the relative pronoun |
| MA | relation MANAGER |
| SEK | relation SECRETARY |
| NOMma | NOM-column of the relation manager, similarly NOMsek, OFma, OFsek, NOMmus |
| x | join operator |
| ; | select operator |
| % | project operator |
| - | subtract |
| = | equal |
| & | and |

The string

((NOMxACC);NOMnom=OFacc)%NOMwh

reads as "join the relations addressed by the nominative and accusative of HAVE, select those tuples where the NOM-column of the relation in the nominative equals the OF-column of the relation in the accusative, and project for printing the NOM-column of those relations where the noun phrases contain interrogatives." In terms of a question against the sample data base this means:

Which manager has which secretary?

NOM, the relation in the nominative, is MA for MANAGER, ACC is SECRETARY: MAxSEK. NOMnom is the NOM-column of MANAGER, OFacc is the OF-column of SECRETARY, the relation in the accusative: NOMma=OFsek. Both nominative and accusative noun phrases contain interrogatives. Therefore the NOM-columns of both must be projected for output: %NOMma,NOMsek.

((MAxSEK);NOMma=OFsek)%NOMma,NOMsek

The following data base operations result from this string: join of the relations MANAGER and SECRETARY:

| MANAGER | | SECRETARY | |
|---|---|---|---|
| NOM | OF | NOM | OF |
| Stern | Moser | Moser | Sauer |
| Stern | Moser | Mahle | König |
| Stern | Moser | Maier | Küfer |
| Stern | Mahle | Moser | Sauer |
| Stern | Mahle | Mahle | König |
| Stern | Mahle | Maier | Küfer |
| Sauer | König | Moser | Sauer |
| Sauer | König | Mahle | König |
| Sauer | König | Maier | Küfer |

In the actual implementation, the select operation precedes the join for reasons of economy. The result is an equi-join, where only those tuples are joined where the equality requested by the select operator exists. This join is shown for the test cases wherever it applies. Similarly, where only one relation is involved, the tuples resulting from the selection operation are shown, not the whole relation. The columns inspected for selection are shown in full. For the present example, the result of the equi-join is the tuple:

```
(   MA     x    SEK  )
Sauer König Moser Sauer
```

The columns inspected for selection are the NOM-column of MANAGER and the OF-column of SECRETARY:

```
NOMma = OFsek
Stern   Sauer
Stern   König
Sauer   Küfer
```

Equality is true for "Sauer". The corresponding data is now projected, Sauer and the secretary of Sauer:

```
NOMma , NOMsek
Sauer   Moser
```

The printed result of the operations and answer to the question

Which manager has which secretary?

then is:

```
MANAGER   SECRETARY
Sauer     Moser
```

In the case of negation in wh-questions and relative clauses, a set thus found is subtracted from the set to be projected, so that "which manager does not have a secretary" is interpreted as "find the managers who have secretaries and subtract them from the set of managers, to get the managers who do not have secretaries". In yes/no questions, also, the positive case is searched in the data base, and the answer depends on whether the result is an empty set. So, "does Moser have no manager" is interpreted as "find the manager of Moser". If the resulting list is empty, there is no manager of Moser and the answer is YES, if it is not, the answer is NO.[2]

The test cases are ordered as follows:

A. No negation
  1. Questions
    a. Two relations: subject and object are nouns
    b. Relation and interrogative pronoun: subject or object is an interrogative pronoun
    c. Relation and name: subject or object is a name
    d. Apposition, two relations, one in the nominative
    e. Apposition, two relations in the accusative
    f. Apposition, three relations
    g. Apposition, name in the accusative
    h. Apposition, name in the nominative
    i. Apposition and two interrogative pronouns the apposition belongs to one of the pronouns
  2. Relative clauses
    a. Two relations
    b. Relation and name
B. Negation
  1. Questions
    a. Two relations
    b. Relation and interrogative pronoun
    c. Relation and name
    d. Appositions
  2. Relative clauses
    a. Two relations
    b. Relation and name

For ease of reference, the actual test cases below are each preceded by their section headings according to the outline above.

### A. No negation

### A.1. Questions

### A.1.a. Two relations

General schema:

((NOMxACC);NOMnom=OFacc)%NOMwh

Welchen Manager hat welche Sekretärin?
Which manager does which secretary have?

```
( ( SEK    x    MA  )  ;NOMsek=OFma)  %NOMma,NOMsek
Moser Sauer Stern Moser  Moser  Moser   Stern Moser
Mahle König Stern Mahle  Mahle  Mahle   Stern Mahle
                         Maier  König
```

Welche Sekretärin hat welcher Manager?
Which secretary does which manager have?

```
( ( MA     x    SEK )  ;NOMma=OFsek)  %NOMsek,NOMma
Sauer König Moser Sauer  Stern Sauer    Moser  Sauer
                         Stern König
                         Sauer Küfer
```

### A.1.b. Relation and interrogative pronoun

General schema:

(ACC;OFacc)%OFacc,NOMacc

Wer hat welche Sekretärin?
Who has which secretary?

```
(   SEK      ;OFsek)  %OFsek,NOMsek
Moser Sauer  Sauer     Sauer Moser
Mahle König  König     König Mahle
Maier Küfer  Küfer     Küfer Maier
```

Wen hat welche Sekretärin?
Whom does which secretary have?

The accusative is not a relation; the answer is not defined.

### A.1.c. Relation and name

General schema:

(ACC;OFacc=NOM)%NOMcc

Welcher Manager hat Moser?
Which manager has Moser?

The accusative is not a relation; the answer is not defined.

[2] DOCH was selected as the answer in GERMAN, because NO confuses those speakers who use NO to answer negated questions in the affirmative.

Welchen Manager hat Moser?
Which manager does Moser have?

```
(    MA      ;OFacc=Moser)  %NOMma
Stern Moser  Moser          Stern
             Mahle
             König
```

**A.1.d. Apposition, two relations, one in the nominative**

General schema:
((NOMxAPP));NOMnom=OFapp)%NOMwh
only for appositions to the accusative

The relation in the apposition takes the place of the relation in the accusative in the scheme for A.1.a. If one of the two relations is in the nominative, and the other in the apposition to the accusative, the formulation "whom as" is equivalent to "which".

Which manager has whom as a secretary?

is equivalent to

Which manager has which secretary?

If the question begins with the accusative, there are two readings: one placing the apposition with the preceding nominative, and a second reading which places it with the accusative. This second reading is the preferred reading. Therefore, the examples

Welcher Manager hat wen als Sekretärin?
Which manager has whom as his secretary?

Wen hat welcher Manager als Sekretärin?
Whom does which manager have as his secretary?

are all translated like:

Welcher Manager hat welche Sekretärin?
Which manager has which secretary?

**A.1.e. Apposition, both relations in the accusative**

General schema:
((ACCxAPP);NOMacc=NOMapp)%OFapp,NOMwh

Wer hat welchen Manager als Sekretärin?
Who has which manager as his secretary?

Welchen Manager hat wer als Sekretärin?
Which manager does who have as his secretary?

```
( (Ma      x   SEK )    ;NOMma=Nomsek)  %OFapp,NOMma
There is no equality     Stern Moser     NONE FOUND
so the join is empty     Stern Mahle
                         Sauer Maier
```

**A.1.f. Apposition, three relations**

General schema:
((NOMxACCxAPP);
NOMnom=OFapp&NOMacc=NOMapp)%OFapp

Welcher Manager hat einen Musiker als Sekretärin?
Which manager has a musician as his secretary?

```
(( MA x MUS x SEK ) ;NOMma=OFsek&NOMmus=NOMsek) %OFsek
Sa. Ko. Mo. Mo. Sa.  Stern Sauer Moser Moser     Sauer
                     Stern König Pahle Mahle
                     Sauer Küfer Peter Maier
```

**A.1.g. Apposition, name in the accusative**

General schema:
(ACC;NOMacc=APP)%OFacc

Wer hat die Sekretärin Moser?
Who has the secretary Moser?

```
(    SEK     ;NOMsek=Moser)  %OFsek
Moser Sauer  Moser            Sauer
             Mahle
             Meier
```

Wer hat Moser als Sekretärin?
Who has Moser as his secretary?

The name and the apposition are permuted by the grammar, so that the input to the translation is the same as in the previous example.

**A.1.h. Apposition, name in the nominative**

General schema:
(APP;OFapp=NOM)%NOMapp
only for appositions to the accusative

Wen hat Sauer als Sekretärin?
Whom does Sauer have as his secretary?

In one reading of this sentence, "as a secretary" is read as apposition to "Sauer", and the question cannot be answered. The second, preferred, reading of the sentence places the apposition with the accusative:

```
(    SEK     ;OFsek=Sauer)  %NOMsek
Moser Sauer  Sauer           Moser
             König
             Küfer
```

Where the name is in the apposition,

Wen hat die Sekretärin Moser?
Whom does the secretary Moser have?

there is no second reading: the answer is not defined.

**A.1.i. Apposition and two interrogative pronouns**

General schema:
(APP;NOMapp)%NOMapp,OFapp
only for appositions to the accusative

Wer hat wen als Sekretärin?
Who has whom as his secretary?

```
(    SEK     ;NOMsek)  %OFsek,NOMsek
Moser Sauer  Moser       Sauer Moser
Mahle König  Mahle       König Mahle
Maier Küfer  Maier       Küfer Maier
```

Wen hat wer als Manager?
Whom does who have as his manager?

Again, the first reading associates "manager" with the noun phrase preceding it, and there is no answer. The second reading places the apposition with "wen" and is translated like the previous example, though with different relations.

```
(    MA      ;NOMma)  %NOMma,OFma
Stern Moser  Stern     Stern Moser
Stern Mahle  Stern     Stern Mahle
Sauer König  Sauer     Sauer König
```

### A.2. Relative clauses

### A.2.a. Two relations

General schema:
((NOMxACC);NOMnom=OFacc)%NOMrel

Manager, der eine Sekretärin hat
Manager, who has a secretary

```
( ( MA      x     SEK )    ;NOMma=OFsek)  %NOMma
Sauer König Moser Sauer  Stern Sauer     Sauer
                         Stern König
                         Sauer Küfer
```

Manager, den eine Sekretärin hat
Manager, whom a secretary has

```
( ( SEK     x     MA   )   ;NOMsek=OFma)  %NOMma
Moser Sauer Stern Moser  Moser  Moser     Stern
Mahle König Stern Mahle  Mahle  Mahle
                         Maier  König
```

### A.2.b. Relation and name

General schema:
(ACC;OFacc=NOM)%NOMrel

Manager, den Moser hat
Manager whom Moser has

```
(    MA     ;OFma=Moser)  %NOMma
Stern Moser  Moser          Stern
             Mahle
             König
```

Manager, der Moser hat
Manager who has Moser

The accusative is not a relation; the answer is not defined.

### B. Negation

### B.1. Questions

### B.1.a. Two relations

General schema:
(NOMwh-(((NOMxACC);NOMnom=OFacc)%NOMwh))

Welcher Manager hat keine Sekretärin?
Which Manager does not have a secretary?

```
(NOMma-(( ( MA      x     Sek)      ;NOMma=OFsek)  %NOMma)      )
Stern    Sauer König Moser Sauer  Stern Sauer               Stern
Stern                             Stern König
Sauer                             Sauer Küfer     Sauer
```

Welchen Manager hat Keine Sekretärin?
Which manager does no secretary have?

```
(NOMma-(( ( SEK     x     MA   )    ;NOMsek=OFma)  %NOMma)      )
Stern    Moser Sauer Stern Moser  Moser  Moser    Stern
Stern    Mahle König Stern Mahle  Mahle  Mahle    Stern
Sauer                             Maier  König              Sauer
```

### B.1.b. Relation and interrogative pronoun

Who does not have a secretary?

The general schema would look like:
(NOMwh-(ACC;OFacc)%NOMacc)

But since the interrogative pronoun does not contain a relation, there is no set to subtract from. These questions cannot be answered in USL, because the set often implied by the context or the meaning of words is not known. The set is given by formulations like "which x does not have y".

### B.1.c. Relation and name

General schema:
NOT(ACC;OFacc=NOM)%NOMacc

Hat Moser keinen Manager?
Does Moser have no Manager?

```
NOT(    MA      ;OFma=Moser)   %NOMma
   Stern Moser  Moser           Stern   the set is
                Mahle                   not empty:
                Maier                      DOCH
```

Hat Moser kein Manager?
Does no manager have Moser?

The accusative is not a relation; the answer is not defined.

### B.1.d. Appositions

Wer hat die Sekretärin Moser nicht?
Who does not have the secretary Moser?

Wer hat keine Sekretärin Moser?
Who has no secretary named Moser?

These examples can be interpreted as "who of the people having secretaries have a secretary other than Moser". This interpretation is not implemented, because for the majority of negated questions containing interrogative pronouns, there is no interpretation in USL. So the exceptions are not interpreted either, to avoid confusion.

The general schema would look like:
(OFacc-((ACC;NOMacc=APP)%OFacc))

The examples above would be translated as:

```
(OFsek-((   SEK     ;NOMsek=Moser)  %OFsek)     )
Sauer  Moser Sauer   Moser            Sauer
König                Mahle                    König
Küfer                Maier                    Küfer
```

### B.2. Relative clauses

### B.2.a. Two relations

General schema:
(NOMrel-(((NOMxACC);NOMnom=OFacc)%NOMrel))

Sekretärin, die keinen Manager hat
Secretary who does not have a manager

```
(NOMsek-(( ( SEK    x    MA  )   ;NOMsek=OFma) %NOMsek)   )
Moser    Moser Sauer Stern Moser  Moser  Moser  Moser
Mahle    Mahle König Stern Mahle  Mahle  Mahle  Mahle
Maier                             Maier  König       Maier
```

Sekretärin, die kein Manager hat
Secretary, whom no manager has

```
(NOMsek-(( ( MA     x    SEK )   ;NOMma=OFsek) %NOMsek)   )
Moser    Sauer König Moser Sauer  Stern Sauer   Moser
Mahle                             Stern König        Mahle
Maier                             Sauer Küfer        Maier
```

### B.2.b. Relation and name

General schema:
(NOMrel-((ACC;OFacc=NOM)%NOMrel))

Sekretärin, die den Sauer nicht hat
Secretary, who does not have Sauer

The accusative is not a relation; the answer is not defined.

Sekretärin, die der Sauer nicht hat
Secretary whom Sauer does not have

```
(NOMsek-((   SEK   ;OFsek=Sauer)  %NOMsek)      )
Moser  Moser Sauer  Sauer          Moser
Mahle                König                 Mahle
Maier                Küfer                 Maier
```

### 5. WITH and WITHOUT

Prepositional phrases containing WITH and WITHOUT can also be related to OF-phrases (Lees 1960:93) and to HAVE (Poldauf 1967:33f.), unless the prepositions are governed by the verb or are part of instrumental, comitative, or adverbial phrases.

Where WITH-phrases are defined as prepositional complements of nouns, verbs or adjectives, they are related to the corresponding WITH-column in the corresponding relation. WITHOUT should then be treated as "not with", but this is not implemented. Phrases containing WITH and WITHOUT that are not complements are interpreted in the following manner: The NOM-column of the relation addressed by the noun governing the prepositional phrase is related to the OF-column of the relation addressed by the prepositional phrase. As a result, except for the columns for projection and subtraction, the following groups of phrases each have the same translation:

(SEK;OFsek=x)
secretary of x
x has secretary
x with secretary
x's secretary

NOT(SEK;OFsek=x)
not secretary of x
x has no secretary
x without secretary
not x's secretary

((MAxSEK);NOMma=OFsek)
secretary of manager
manager has secretary
manager with secretary
manager's secretary

NOT((MAxSEK);NOMma=OFsek)
not secretary of manager
manager has no secretary
manager without secretary
not manager's secretary

### 6. Problems

In conclusion, some of the problems of this interpretation must be pointed out. If there are relations POSSESS or OWN in the data base, HAVE sentences could be meant to refer to these relations. But there is no mechanism that automatically would look for such relations and either access them directly or resort to them if the standard interpretation fails. A solution to this problem is for the user to define HAVE as a synonym for POSSESS. He can then delete the system-defined HAVE altogether, if he is sure to use HAVE only in the sense of OWN. But he will probably want his definition in addition to the system defined HAVE. This leads to two parses and interpretations. In many cases only one of them will bring results, but sometimes both lead to the same answer. Thus,

Does John have a secretary?

will be interpreted both as:

Is there a secretary of John?

and

Does John own a secretary?

where the second interpretation will fail. But the query

Does John have a car?

can bring answers to both interpretations, depending on the structure of the data base.

A consequence of not having a relation HAVE on the one hand and on the other of expecting all adverbials and prepositional phrases to refer to columns in relations causes difficulties where prepositions are used in conjunction with HAVE and also with BE, as in questions like:

Where does Peter have his office?
To whom is Peter married?
When is Peter's birthday?
Is Peter's birthday on Friday?

The German equivalent of the last two questions uses the verb HAVE:

Wann hat Peter Geburtstag?
Hat Peter am Freitag Geburtstag?

For sentences with HAVE and BE as the main verb, the adverbials and prepositional phrases which have been attached to the verb by the grammar are relocated to the noun phrase addressing the relation that contains the corresponding columns. For the examples above, this means the columns LA, P=to, and TA, respectively, in the relations OFFICE, BIRTHDAY, and MARRIED.

A serious problem for which we currently have no solution results from the fact that there is no interpretation where the accusative does not contain a relation. As can be seen from the examples, it is often intuitively clear what the interpretation of such sentences could be. "The secretary who has Sauer" could easily be the secretary of Sauer in the context of "who has whom to work for", but in the context of "who has which manager", the "secretary who has Sauer" can be the one whose manager is Sauer. As was shown above, the predicate HAVE is a derived predicate, it is applicable only where a more basic association exists between the elements in question. This predicate is found explicitly expressed, where HAVE sentences are expanded by AS-complements.

Peter has a musician as secretary

If there is no AS-complement and the accusative of HAVE contains a common noun, that noun generally points to the predicate; the AS-complement would be redundant.

Peter has a musician

can be read as

Peter has a musician as his musician

But in the proper context, the base predicate can be clearly something else: In a conversation about hobbies of secretaries, the same sentence "Peter has a musician" can mean

Peter has a musician as his secretary

In human dialogue the general rule that the base predicate appears in the accusative noun phrase can be overridden by special contexts, but in the general case the rule holds and can be used in the framework of data base interaction. In the absence of reference to a predicate in the accusative of HAVE, the base predicate can only be deduced from the context. It can be one of the properties of the element in the accusative, e.g. Sauer's being a manager, a father, or an employer, or it can coincide with the predicate referred to by the nominative. Therefore, the attempt at interpretation of such sentences would lead to choosing among plausible alternatives, choices that would remain arbitrary even if carefully made, and the results would be unreliable. We have avoided this at the cost of not providing general interpretations for these cases, even though the individual case is often intuitively interpretable, because we feel that in the framework of data base interaction it is more important for the system to react consistently and reliably than to simulate human dialogue.

Since this paper was originally written, the USL System has moved to using System R as the data base management system, with SQL as the query language and target language for the interpretation. The PRTV-version of USL has been used by small groups in their applications, and initial results are encouraging (Lehmann et al., 1978, Krause, 1979). Further study is necessary, particularly with respect to data-base design and vocabulary definition by users.

## References

Bach, E. (1967): "*Have* and *be* in English syntax," *Language 43,* 462-485.

Bennet, M.R. (1974): "Some extensions of a montague fragment of English," *UCLA Dissertation,* Indiana Linguistics Club (1975).

Bierwisch, M. (1965): "Eine Hierarchie syntaktisch-semantischer Merkmale," *Studia Grammatika V,* 29-86.

Codd, E.F. (1972): "Relational completeness of data base sublanguages," *Research Report RJ 987,* IBM Research Laboratory, San Jose, California.

Cresswell, M.J. (1973): *Logics and Languages.* London.

Dostert, B.H., and Thompson, F.B. (1971): "The syntax of REL English," *REL Report No. 1,* California Institute of Technology, Pasadena.

Kay, M. (1967): "Experiments with a powerful parser," *Second International Conference on Computational Linguistics,* Grenoble, August 1967,

Krause, J. (1979): "Preliminary results of a user study with the 'User Specialty Languages' System and consequences for the architecture of natural language interfaces," *Technical Report TR 79.04.003,* IBM Heidelberg Scientific Center.

Langacker, R.W. (1975): "Functional stratigraphy," *Papers from the Parasession on Functionalism,* April 17, 1975, (ed. R.E. Grossmann, L.J. San, and T.J. Vance), Chicago Linguistic Society, Chicago.

Lees, R.B. (1960): *The Grammar of English Nominalizations,* The Hague.

Lehmann, H., Ott, N., and Zoeppritz, M. (1977): "Language facilities of USL/German, Version III," *Technical Note TN 77.04,* IBM Heidelberg Scientific Center, 1977.

Lehmann, H., Ott N., and Zoeppritz, M. (1978): "User experiments with natural language for data base access," *Proceedings 7th International Conference on Computational Linguistics,* Bergen, Norway, August 1978.

Lehmann, H. (1978): "Interpretation of natural language in an information system," *IBM Journal of Research and Development 22, 5,* 560-572.

Ott, N. (1977): "Interpretation of questions with quantifiers and negation in the USL System," *Technical Report TN 77.10.005,* IBM Heidelberg Scientific Center.

Ott, N. (1979): "Das experimentelle, auf natürlicher Sprache basierende Informationssystem USL," *Nachrichten fuer Dokumentation 3,* 129-139.

Pitha, P. (1971): "Remarks on Possessivity," *Prague Bulletin of Mathematical Linguistics 16,* 33-46.

Pitha, P. (1972): "Remarks on Possessivity II," *Prague Bulletin of Mathematical Linguistics 17,* 25-36.

Poldauf, I. (1967): "The *have* construction," *Prague Studies in English 12,* 40.

Thompson, F.B., Lockemann, P.C., Dostert, B.H., and Deverill, R.S. (1969): "REL: A rapidly extensible language system," *Proceedings 24th National ACM Conference,* New York, August 1969.

Todd, S.J.P. (1975): "PRTV - A technical overview," *Technical Report UKSC 0075,* IBM Scientific Center, Peterlee, England, (also in *IBM Systems Journal 15,* 285-308).

*Magdalena Zoeppritz is a staff member at the IBM Heidelberg Scientific Center. She received the Master of Arts degree in English and general linguistics from the University of Hamburg in 1968.*