

kansas
working
papers
in
linguistics

volume 11

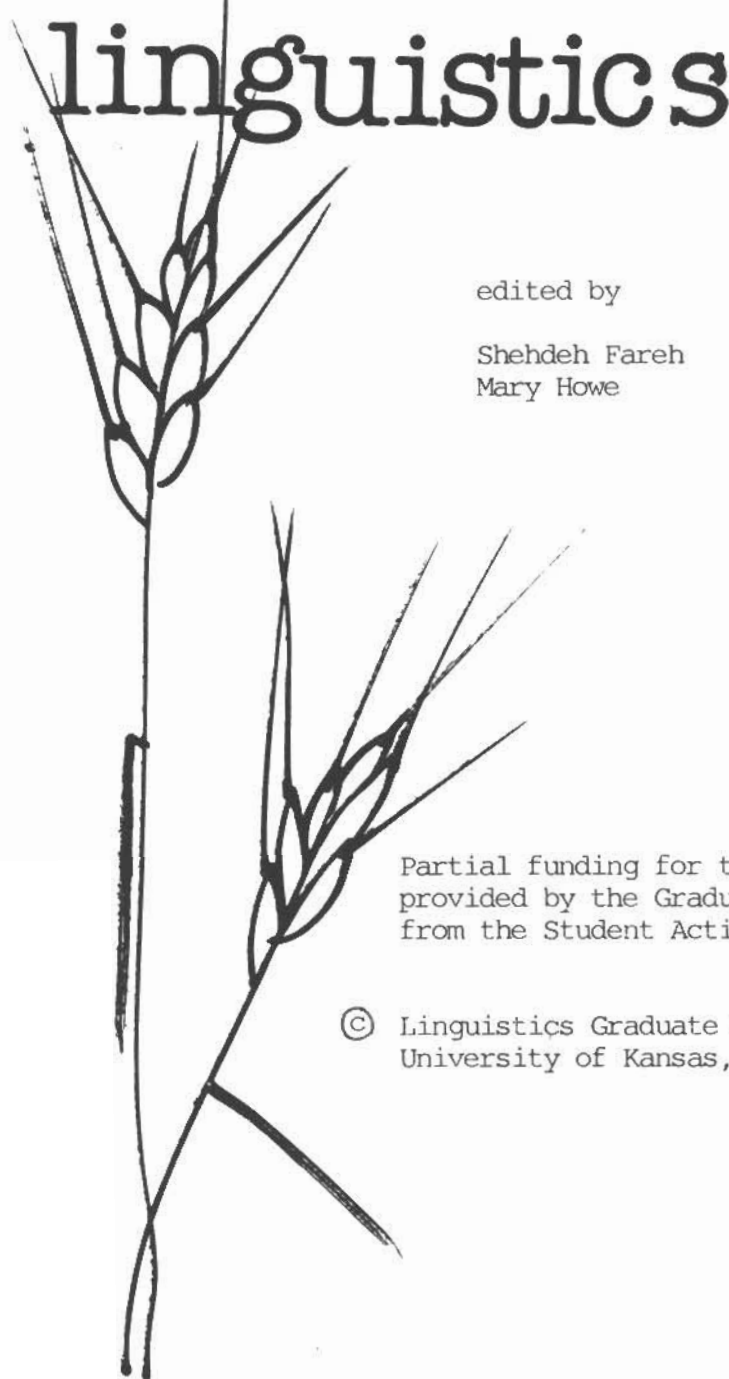
1986

edited by

Shehdeh Fareh
Mary Howe

Partial funding for this journal is
provided by the Graduate Student Council
from the Student Activity Fee.

© Linguistics Graduate Student Association
University of Kansas, 1986



Kansas Working Papers in Linguistics is a regular publication of the Linguistics Graduate Student Association, Department of Linguistics, University of Kansas, Lawrence, KS 66045

Linguistics Graduate Student Association Officers,
1985-1986:

President:	Antonia Folarin
Secretary:	Misaki Shimada
Treasurer:	Charles Schleicher

Aim: *Kansas Working Papers in Linguistics* is intended as a forum for the presentation, in print, of the latest original research by the faculty and students of the Department of Linguistics and other related departments at the University of Kansas. Papers contributed by persons not associated with the University of Kansas are also welcome. The papers published in *KWPL* may not be reproduced without written permission from the Linguistics Graduate Association.

Send all manuscripts and inquiries to:

Editors, *KWPL*
Department of Linguistics
University of Kansas
Lawrence, KS 66045 USA

Requests for individual volumes should be addressed to Linguistics Graduate Student Association at the above address. Institutions producing a similar publication may request a reciprocal agreement.

The cost per issue for Volumes 1 through 6 and Volume 8, number 1, is US\$4.50 postpaid. The cost for Volume 7, Volume 8, number 2, Volume 10, number 1 and Volume 11 is US\$7.50 postpaid. The cost for Volume 9 and Volume 10, number 2 is US\$10.00 postpaid. Reprints of individual articles may be requested for US\$1.50 postpaid. For orders outside the United States and Canada, please add US\$1.50 per volume to help defray the costs of postage (a cumulative index to volumes 1-11 can be found at the back of this issue).

We would like to thank the faculty of the Linguistics Department and the Graduate Student Council for their continuing encouragement and support.

Kansas Working Papers in Linguistics

Volume 11, 1986

<i>Julie Bruch</i>	Expressive Phonemes in Japanese	1
<i>Hiroshi Nara</i>	Approaches to Anaphora Resolution in a Natural Language Database Management System	9
<i>Elzbieta Moszczak</i>	Major Address Patterns in Polish and How They Compare with Major Address Patterns in English	27
<i>Shehdeh Fareh</i>	How Far Does the English Teachers' Pronunciation Affect that of Their Students in Jordanian Public Schools?	45
<i>Andrew Pawley</i>	On Speech Formulas and Linguistic Competence	57
<i>Natasha Beery</i>	I Can't Help Myself: A Lexicalized Expression Examined	89
<i>Mary Howe</i>	Conflicting Speech Formulas: About To and Not About To	101
Cumulative contents of Volumes 1-11		114

APPROACHES TO ANAPHORA RESOLUTION IN NATURAL LANGUAGE DATABASE
MANAGEMENT SYSTEM*

Hiroshi Nara

Abstract: This paper describes in detail the computational process of finding the non-indexical linguistic objects to which pronoun anaphoras resolve. The successful development of discrete steps for anaphora resolution is critical for any natural language interface to a variety of systems. The types of information available for resolution are first identified and the past approaches to resolution for each type are critically reviewed. This paper then describes how anaphora resolution is implemented in the present system using all types of information, including the discourse information of focus, background, and focus shift.

INTRODUCTION

A robust natural language system like the one under development must be able to handle certain cases of indexical expressions without excessive overhead to the entire system. Both spoken and written language makes extensive use of indexicals, including pronouns (SHE, THEM, ITS, and the like), place and time adverbs and adverbials (THERE, NOW, THEN, etc.), determiners and demonstratives of various kinds (THE, WHICH, THOSE, etc.), and other cohesive devices to point back to previous constituents and concepts in a sentence and a discourse.

The area of linguistics in focus is pragmatics. As such, as in any other field of pragmatics, rigorous approaches to indexicals are notoriously difficult, if not impossible, to achieve. This is not only because of the sheer immensity of pragmatic factors that control the syntactic behavior of indexicals but also because of the difficulty in isolating those factors into discrete sets of rules.

In a natural language understanding system (NLU system), the discourse context is established by the interaction between the system and the user; as such, the system must be equipped with certain skills and strategies for processing indexical expressions correctly. This kind of mechanism requires a very well-defined and discrete set of steps to determine what those indexicals mean to the NLU system. Simply put, the problem for the system is to come up with a procedure which determines the meaning of the indexical expressions. It is also desirable for such a procedure to be small and computationally efficient. Given that pragmatic principles are extremely difficult to formalize, it is therefore wiser to use other types of information first for anaphora resolution (i.e., semantic, morphological, syntactic), and then use pragmatic principles as needs arise.

An additional, and closely related, issue is the resolution of non-indexical NPs. Some NPs, like the definite description NPs, behave like indexical expressions in that they often make reference to already-mentioned or otherwise understood entities. The use of the determiner THE for referring back to an indefinite NP in earlier discourse is a typical example.

ANAPHORA RESOLUTION

A good place to begin is the resolution of anaphoric expressions, especially pronoun anaphora. It is the treatment of this type of anaphora that will be the topic of the following discussion. The phenomenon is widespread and poses the kind of problems that must be solved for any other kind of indexicals.

To illustrate these problems, examine the following interaction between the user and the system.

- (1) a. >>DO YOU HAVE SMITH'S ADDRESS?
 b. 911 SUNSET DRIVE
 c. >>IS HE ENROLLED THIS SEMESTER?
 d. NO
 e. >>OK. GET JONES'S ADDRESS
 f. 8 KENTUCKY COURT
 g. >>LIST THEIR PHONE NUMBERS
 h. SMITH 841 3076
 JONES 843 1635
 i. >>ARE THEY BOTH LAWRENCE NUMBERS?
 j. YES
 k. >>WHAT ARE THEIR MAJORS?
 l. SMITH'S MAJOR IS LINGUISTICS
 JONES'S MAJOR IS PHILOSOPHY

In (1a), YOU must be resolved to refer to the system, and this is relatively simple. In (1c), HE points back to the name SMITH. THEIR in (1g) must point to SMITH and JONES, while THEY in (1i) points back to the two phone numbers. In (1k) the anaphora link is made back to SMITH and JONES, not the phone numbers.

There are slightly more complex cases, as illustrated by the following additional examples:

- (2) a. >>GET SMITH'S GPA
 b. SMITH'S GPA IS 3.24
 c. >>WHAT IS IT FOR JONES?
 d. JONES'S GPA IS 2.94
 e. >>HIS MAJOR ADVISOR?
 f. HIS MAJOR ADVISOR IS MCDONALD
 g. >>WHAT IS HIS PHONE NUMBER?
 h. 864 3450

IT in (2c) is an interesting case (the case of lazy pronouns, see Geach 1962, Karttunen 1977), since IT points to GPA, not Smith's GPA.

In (2e), HIS should resolve to JONES, the most recently mentioned name. The same HIS in (2g), however, should now point to MCDONALD, not JONES, as queued by the answer to the preceding query (2f). It may also be noticed that (2g) is ambiguous between WHAT IS HIS (MCDONALD'S) PHONE NUMBER?, and WHAT IS HIS (JONES'S) PHONE NUMBER?.

Listed below are more examples in which pronouns are perhaps more difficult to resolve.

- (3)
- a. >>DID EACH STUDENT TAKE LINGUISTICS 320?
 - b. YES
 - c. >>DID THEY ALL PASS?
 - d. NO, 1 FAILED
 - e. >>DID ANYONE FAIL LINGUISTICS 104 IN 1983?
 - f. YES, 3 FAILED THE COURSE
 - g. >>WHEN DID IT HAPPEN?
 - h. FALL 1983
 - i. >>WAS IT MARY SMITH WHO FAILED LINGUISTICS 320?
 - j. YES

Notice, in (3c), that THEY and EACH STUDENT both refer to the same set of entities, although their grammatical property of NUMBER differs. (3g) exemplifies IT referring to an event, and an event is not always a clearly discernable linguistic entity. See Stirling (1985:20) for further examples. Lastly, (3i) illustrates the use of IT in a cleft construction. and here IT is a dummy NP required for this linguistic device.

Other interesting instances of pronoun behavior include the ones listed below (Tennant 1981:120).

- (4)
- | | |
|------|--|
| YOU | You never know what will happen tomorrow |
| WE | We don't do that here |
| THEY | They dug up the road today |
| IT | It's raining |
| IT | It's time for another beer |
| IT | It was utter chaos at the PTA meeting |
| IT | How is it spelled? |
| HE | Bill tickled John and he squirmed. |
| | (problem of control) |

Despite the diversity of the data, it is not difficult to resolve a sizeable portion of pronoun anaphora by simply adopting a mechanism which keeps track of various types of information explicitly present in the interaction between the system and the user. In the next section. these various types of information for anaphora resolution will be investigated.

SOME TYPES OF INFORMATION AVAILABLE FOR ANAPHORA RESOLUTION

At least four different types of information are available: they are morphological, syntactic, semantic, and pragmatic information. Each type can be extracted from the sentence or the discourse to resolve pronoun anaphors.

At the morphological level, the grammatical categories of number and gender can often provide sufficient information to resolve the anaphor (NOTE 1).

Rather unsophisticated syntactic information, such as the case role of a pronoun in a sentence, is known to provide the information which can often be used for the resolution of the pronoun. For instance, Hajicova & Sgall (1985:266) reports that if two NPs are possible resolutions of a pronoun, then the pronoun is more likely to resolve to the subject NP; otherwise it is more likely to resolve to the objective NP than other oblique NPs.

This observation is consistent with the English topical structure (and that of many other languages) that the topical element tends to be preposed in a sentence. Similar studies contained in much of the computational linguistics literature in this area are applications of well-known linguistic studies of pronouns in English as in Lasnik 1976, Wasow 1979, Chomsky 1980, etc.

These earlier investigations contributed to a more sophisticated method of extracting syntactic information from the input sentence using the Government and Binding Theory. This approach, as summarized below, can be used very profitably in anaphora resolution.

Chomsky recognizes three distinct types of NP: anaphors, pronominals, and lexical NPs. In his framework, anaphors include reflexive anaphors, such as MYSELF, EACH OTHER, and pronominals including what are normally called pronouns. In the EST, reflexive anaphors and pronominals are not generated in the base, but the NPs meeting certain syntactic and semantic restrictions are transformationally pronominalized. Thus the structure such as $[[john]_{np}[[hit]_{v}[john]_{np}]_{vp}]_s$ is realized as 'John hit himself' on the surface.

In the more current version of TG, the Government and Binding Theory, the Base generates a class of structures, called D-Structures, which are then mapped onto S-Structures via the Transformational Component and the Case-marking Component. The pronominals are listed in the lexicon and, as such, they are directly generated by the Base. The S-Structures then become the inputs of two other parallel components: the one which maps the S-Structures onto the Surface Structures, and the other which maps them onto semantic representations (Logical Form).

A semantic rule of interest contained in this second component is a rule called Indexing Rule, which marks all NPs in the input structure with integer indices that are not necessarily distinct.

(The idea behind all this is that eventually all coreferring NPs will have the same index. so that the coreference relation among NPs can be accounted for.) In order to remove impossible interpretations, a matching rule is applied whereby all the NPs having the same index but different features (e.g. gender, number, person, etc.) are removed.

The matching rule still fails to remove all impossible interpretations. including the ones listed below (Radford 1981:367).

- (5) John₂ hurt himself₃.
- (6) John₂ hurt him₂.
- (7) John₂ hurt Fred₂.

Faced with the problem, Chomsky proposed three binding conditions designed to remove those inadequacies illustrated above:

- (8) Binding Conditions
 - a. an anaphor is bound within its minimal governing category, if it has one.
 - b. a pronominal NP must be free in its minimal governing category. if it has one,
 - c. a lexical NP must be free everywhere.

For the purpose of brief illustration of these Binding Conditions, assume the structure in (9) for (5)-(7) (Radford 1981:368).

(9)

	S-BAR				
COMP			S		
-WH		NP2	TENSE		VP
e		JOHN	+PAST	V	NP2/3
			e	HURT	HIMSELF HIM FRED

Assume further that the last NP node is occupied by HIMSELF₃. HIMSELF₃ in (9) is GOVERNED by the verb HURT, and since the NP node or an S node containing the verb is S, the S node is the GOVERNING CATEGORY for the NP HIMSELF₃. Due to (8a), HIMSELF points to HIMSELF since NP2 C-COMMANDs NP3. If the Objective NP2 node is filled with HIM instead, then JOHN can no longer serve as the antecedent of HIM, as stipulated by (8b). Although (7) is a possible outcome of indexings, JOHN and FRED cannot be co-indexed in accordance with (8c).

In addition to these simple examples. the GB Theory seems to make correct predictions in many complex cases involving anaphors (see Berwick & Weinberg 1984 for extensive examples). The GB approach to anaphora has been implemented by some computational linguists (Guenther & Lehmann 1984a, 1984b) with good results. It

is easy to see, then, that syntactic information of various kinds, both simple and sophisticated, can be profitably used for intrasentential pronoun resolution.

A serious complication with this approach in computational terms is the fact that not all NLU system parsers are GB Theory compatible, including the present system. Being compatible with the GB Theory means implementing the transformational component, and certain factors relating to anaphora resolution are direct consequences of transformations (such as movement traces, which can nevertheless serve as an antecedent of an anaphor). Another concern is that the resolution is very computation-intensive as it must manipulate large data structures in addition to a large amount of computation required for establishing c-commanding and governing relations among the NPs and the governing categories for those NPs.

Certain heuristic strategies can be adopted to emulate some of these useful observations in a more efficient manner. In addition to the fact that these strategies are more efficient, they can also be implemented in non-GB parsers relatively easily.

The general approach can be outlined as follows. First, if the anaphor is reflexive, a compatible NP is sought in the clause/sentence of the same embedding level (cf. (8a)). Non-reflexive anaphors can be resolved using the following procedure: if the sentence contains no clauses, a non-reflexive pronoun seeks its antecedent to its left and resolves to the first feature-compatible one on the same embedding level. If the sentence involves embedding, an anaphoric NP at the embedding level i resolves to the first compatible NP on the level j ($i \neq j$) to the left of the anaphor. Compare this general strategy with (8b). Lastly, (8c) deals with lexical NPs and involves names like JOHN, MARY, and other rigid designators which resolve to themselves, and other NPs such as THE STUDENT WITH THE HIGHEST GPA, A STUDENT WHO ..., etc. Although these NPs must also be resolved, it does not directly involve anaphora in the sense discussed here.

At the semantic level, sortal information (selectional restrictions) associated with each NP can be profitably used for resolution. In (lc), HE resolves to SMITH, almost by default, but if SMITH is marked MASCULINE, the case for resolving this way is further strengthened. Sortal information is also used in resolving THEY and THEIR in (li) and (lk), respectively. In general, however, determining sortal correctness is very computation-intensive and may not be suited in some cases. These features are particularly useful for the analysis of highly inflecting languages (Karttunen 1984). In connection with this topic, Comrie (1981) lists a number of grammaticalized sortal categories that are commonly used by a number of natural languages, including ANIMACY and HUMAN, and those parameters can also be incorporated for writing a more computationally efficient algorithm.

When resolution involves the crossing of a sentential boundary, some of the controlling factors in the behaviors of pronouns become

less generalizable. In this area, the majority of studies comes from cognitive science. For instance, a number of studies (Groc 1977, Groc, Joshi, & Weinstein 1993, Sidner 1979, Sidner 1983, Reicheman 1978, Reicheman 1984) suggest that the human memory structures can be classified into two types: short-term and long-term. Short-term memory is said to have a small storage capacity but allows a faster information retrieval, whereas long-term memory has a large capacity but slower retrieval time. It seems that intersentential pronouns are used when referring to objects in short-term memory, and other types of NPs are used when referring to those in long-term memory. A topic shift is signaled by the failure to resolve the NP in short-term memory. In this case, the NP is most likely resolved in long-term memory, and if it is, the object to which the NP resolves will be reinstated in short-term memory.

These and some additional studies (Carpenter & Just 1977, Clark and Sengul 1979, Lesgold, Roth & Curtis 1979, Chang 1980, McKoon & Ratcliff 1980, Guindon 1985) offer cognitive evidence as to how short-term memory might serve in the resolution of anaphora. It is argued that the objects to which pronouns may point are chosen from among the objects in short-term memory (focus), and other anaphors are resolved in long-term memory (operating memory).

Examine the following example for the purpose of understanding how this mechanism of focus and operating memory may be implemented.

(10)

INTERACTION

- a. >>WHAT DEGREE PROGRAM IS SMITH IN?
- b. MA IN LINGUISTICS
- c. >>WHEN WAS HE ADMITTED IN THE PROGRAM?
- d. HE WAS ADMITTED IN THE PROGRAM IN F84
- e. >>IS THERE A WOMAN WHO RECEIVED A PHD IN 1982?
- f. YES, JANE MCALESTER
- g. >>WHAT WAS IT IN?
- h. PHILOSOPHY
- i. >>DO YOU HAVE HER CURRENT ADDRESS
- j. 1824 MICHIGAN ST

ANAPHORA LIST

- k. ((SMITH)(PROGRAM)) after (10a)
- l. ((MA IN LING)(SMITH)) after (10b)
- m. ((WOMAN)(PHD)(SMITH)(MA IN LING)) after (10e)
- n. ((JANE MCALESTER)(PHD)(SMITH)(MA IN LING)) after (10f)
- o. ((JANE MCALESTER)(PHD IN PHILOSOPHY)
(SMITH)(MA IN LING)) after (10h)
- p. ((1824 MICHIGAN)(JANE MCALESTER)
(PHD IN PHILOSOPHY)(SMITH)(MA IN LING)) after (10j)

Supposing that (10a) is the first query, the word SMITH and DEGREE PROGRAM are kept in a list in order of occurrence, something like (10k), keeping the most recent one in the left-most position. HE in (10c) is resolved to SMITH by checking each entry in (10k) from the left, since HE and SMITH agree in NUMBER and GENDER. After (10b), the

list is updated to look like (10l) by placing MA IN LING, which is consistent with the most-recent-one-first policy.

In (10d), HE similarly resolves again to SMITH. When (10e) is processed, the NPs are appended to the left of the list, updating it to a list like (10m). The answer (10f) replaces the first item in the list since A WOMAN WHO ... and JANE MCALESTER are coreferential. The updated list now appears as (10n).

It is easy to see that IT in (10g) now resolves to PHD, updating the list to ((JANE MCALESTER)(PHD IN PHILOSOPHY)(SMITH)(MA IN LING)), as in (10o). HER CURRENT ADDRESS is now pushed down in the list in (10i), which is later replaced by 1824 MICHIGAN ST when (10j) is given as the answer to the query. The final list will have the form ((1824 MICHIGAN ST)(JANE MCALESTER) (PHD IN PHILOSOPHY)(SMITH)(MA IN LING)).

Suppose the next query contained IT, SHE, and HE. We shall follow the same procedure characterized above to see whether it gives us the right result. IT resolves to the first item in the list with correct grammatical and morphological properties, namely 1824 MICHIGAN ST. It can also resolve to PHD IN PHILOSOPHY, perhaps less plausibly. SHE resolves to JANE MCALESTER, and HE, SMITH. The results are in accordance with our understanding of the workings of these pronouns, and, as far as these examples are concerned, the procedure appears to work well.

As the interaction time with the system increases, the list of this form becomes longer. There is no inherent limit to the length of the list. Hobbs reports (1976) that 85 percent of pronoun anaphoras are resolved within two previous sentences using only the NUMBER, GENDER, and syntactic information. In one study reported in Hobbs 1976, 98 percent of the anaphoras could be resolved within the current and the previous sentence.

If Miller is correct in saying that short-term memory can store only up to about nine 'units' (Miller 1956), then it does not seem necessary to keep track of inordinate number of items in a possible anaphor list as above for resolution. It will probably suffice to have about 10 units in the active list (i.e., the focal list), and the rest may be placed in a more permanent memory structure.

ANAPHORA RESOLUTION AND EXTENSION

Suppose that John Doe is the only man student majoring in linguistics. I can unambiguously refer to the physical entity by using a definite description THE MAN WHO MAJORS IN LINGUISTICS. The idea behind the definite description is that, given the frame of reference (indices), it is possible to uniquely identify the person using just the description so given. A proper noun like JOHN DOE is a rigid designator, and, as such, it is capable of picking out just the person from a pool of possible candidates. In the case at hand, the definite description THE MAN WHO ... and JOHN DOE are co-

referential and co-extensional. They are simply two different ways of getting at the same individual. The definite description and JOHN DOE are said to co-refer to the same object John Doe in the informal model specified earlier.

As clearly seen from our example, co-reference and extension are semantic notions and in the computational context they are not directly related to pronoun resolution. Just as it cannot be said that the semantics of a pronoun is complete by simply assigning the pronoun to another NP, the meaning of a pronoun to an NLU system is not fully given by resolving the pronoun to a non-anaphoric NP. In addition to finding a non-anaphoric NP, the system must also make a connection between the NP so resolved and some kind of semantic object it refers to.

This cannot be done, of course, without first determining which LINGUISTIC OBJECT (i.e., constituent) a particular pronoun can point to. In this sense, pronouns are removed from semantic objects twice, intervened by other denoting phrases and words.

The following schematic representation might be helpful here.

(11)

LINGUISTIC OBJECTS		----->	SEMANTIC OBJECTS
PRONOUN	OTHER NPS		
HE --->	JOHN		X
---	BILL		Y
---	THE MAN WHO ...		Z
	...		

What has been discussed so far is various approaches of resolving pronoun anaphora, that is, the method of connecting in some unambiguous manner the two types of linguistic objects-- pronouns and other NPs. To reiterate, resolution is the process of finding the correct relation between pronouns and co-referring NPs, whereas reference is that relation between linguistic objects and their semantic extensions.

In the physical world, the extensions of rigid designators such as JOHN and BILL and the definite descriptions such as THE MAN WHO ... are the physical entities, or from a slightly different point of view, the set of properties which uniquely constitute these individuals.

In relational database systems, the semantic object denoted by a proper name is often that tuple from a relation (or a join of relations) keyed with JOHN (since JOHN is the primary key as it is a rigid designator), and there is exactly one tuple keyed by JOHN. Recall also that each tuple is an (ordered) set of attribute values (i.e., properties), and this is consistent with formal linguistic semantics.

Not all the extensions of NPs are simply tuples, and the kind of extension a particular NP takes depends largely on the type of database organization. One of the advantages of the relational database can be clearly seen here, since relational databases can limit the number of access paths to the correct semantic extension.

Several types of semantic objects in a relational database denoted by NPs can be readily identified. The first type, SIMPLEX DATA OBJECT, is the data value extracted from the database by using a combination of the standard relational algebra operators (JOIN, PROJECT, and SELECT). For example, queries such as GET MARY'S ADDRESS against (12) is processed by first finding the tuple from a table in which MARY appears as the primary key for that table. The field named ADDRESS is accessed and returned as the value of that access function.

```
(12)
STUDENT
  NAME          ID          ADDRESS          PHONE
MARY SMITH     111222    1824 MICHIGAN    843-1111
```

The second type may be called the COMPLEX DATA OBJECT. This type of data object requires other operations other than those mentioned above. One accessing mechanism of the COMPLEX DATA OBJECT can be illustrated by the query: GET MARY'S GPA, and assume that there is no such field GPA in the tuple accessed with the primary key MARY. Mary's GPA can be computed in the well-known manner, if the system had access to the list of courses, credit hours, and the grades she made in these courses, such as the one illustrated in (13). In the retrieval of complex data objects, therefore, simplex data objects are retrieved first, then necessary computations are performed on the result of the simplex data objects.

```
(13)
MARY SMITH
  SEMESTER      COURSENAME      CR      GRADE
F85             LING 700         3        A
F85             LING 720         3        B
F85             PHIL 320         3        A
...
```

```
TCREDIT:=0;TPTS:=0;
WHILE NOT END OF TABLE
  BEGIN
    TCREDIT := TCREDIT+CR;
    CASE GRADE OF
      A: PTS:=CR*4;
      B: PTS:=CR*3;
      ...
    TPTS:=TPTS+PTS
  END;
GPA:=TPTS/TCREDIT;
```

Conceived in this way, then, the task of the system is to define a function to access the correct data objects in the database, and apply that function to the correct domain. The value of the function is the data value sought. The access function thus defined to retrieve the correct data objects constitutes the other half of the semantics of a query.

CONFIGURATION OF ANAPHORA STACK

The information necessary for the resolution of anaphora must be kept in an accessible and well-organized manner. In an earlier section of this paper (refer to (10)), one method of keeping track of anaphoric information was presented. I consider the approach to be basically correct, and I will show below how this approach can account for a good portion of pronoun anaphora.

Any system designed to resolve anaphors must appeal to some type of data structure which keeps track of which NPs are in focus, and which are in the background at any given time. In addition, it should be noted that the focus is a dynamic data structure and as such it must be updated constantly. Recall also that the most recent NPs (i.e., those in focus) must be made readily available since these NPs are most likely to be accessed in anaphora resolution. The kind of ordering of the information units in the data structure has to do with the flow of discourse, and the ordering is updated constantly, as new NPs shift into and out of focus.

The most efficient storage structure for this type of data is the anaphora data stack (ASTACK). It consists of three different types of storage, and their types differ according to the purpose of the information they store. The first type stores the information for intrasentential anaphora resolution. This ASTACK, called Sentential ASTACK (SASTACK), contains morphological, sortal, syntactic, and other information on all NPs appearing in the sentence in question.

The second type is called Discourse ASTACK (DASTACK) and it has the discourse-related information. The DASTACK is a set of ASTACKs, arranged in the order of topical and focal strength at any given time.

The content of the SASTACK is first incorporated into the DASTACK and then emptied after each sentence is processed, but DASTACKs are updated whenever a change in focus is detected. Updating DASTACKs means that the focally strongest ASTACK is brought up from its original position in the stack to the top, unless it is already there. In the following illustration, the DASTACK at a time point i is updated to resemble $DASTACK_{i+1}$, after a match was found with NP3. NP3 now moves to the top of the DASTACK since it is now in focus.

(14)	DASTACK _i		DASTACK _{i+1}
top	NP1		NP3
	NP2	update	NP1
	* NP3	----->	NP2
	.		.
	.		.
	.		.

There are few constraints on the depth of the stack, but if the results of the studies mentioned earlier in connection with the amount of information held in the short-term memory structure is correct, a depth of 10 should be adequate. The depth can change easily to accommodate differing requirements.

If the number of stack elements in the DASTACK exceeds ten, the bottom element which will get pushed out. It will be placed on a long-term memory structure called Background ASTACK, or BASTACK. It contains the similar types of information as in the DASTACK. BASTACKs are accessed rarely, only when a pronoun is not resolved in the SASTACK or DASTACK.

When the parser encounters an instance of NP, it concatenates the actual NP, the information extracted from the NP and the surrounding environment, and its referent (if possible) into a SASTACK element. The SASTACK contains as many elements as there are constituents that can be pointed to by anaphoras in the sentence, keeping the most topical (i.e., recent) one on top if such information is available. If any one of these constituents encountered by the parser is anaphoric, then the topmost element in the SASTACK is checked to determine whether the anaphor can point to it. The decision depends upon the information available at that time as noted above. If the test fails, the next SASTACK element is checked for compatibility. The procedure stops when a match is found or the bottom of the stack is reached.

An anaphor is replaced by a non-anaphoric NP, if a match for the anaphor is found, so that the access function in the database management language could later be generated without anaphors, making the function expression directly executable.

If all the anaphors are resolved within the current sentence, the content of the SASTACK is pushed onto the DASTACK. At this time, any duplicates in the DASTACK this pushdown might have created will be eliminated. In addition, if any DASTACK element is pushed out from the bottom, it is then added to the BASTACK. If the anaphor does not resolve within the sentence, the search is conducted through the DASTACK for a match.

Suppose that a match is found in the DASTACK. The anaphor then resolves to that element in the DASTACK, causing, at the same time, the element to which the anaphor resolved to rise to the top of the DASTACK. This will place the NP in the strongest focus, and updates the entire DASTACK. On the other hand, if the search is not successful in the DASTACK, the search continues in the BASTACK, until it

fails (a case of nonresolution) or until a match is found. If successful, the matched NP will be brought onto DASTACK (into focus), and all the original DASTACK elements will be displaced by one.

The process is more simply represented in an algorithmic form below:

```
(15)
FOR EACH NP IN THE INPUT SENTENCE
  IF THE NP IS NON-ANAPHORIC
    PUSH IT AND AVAILABLE INFORMATION ONTO THE SASTACK
  IF THE NP IS ANAPHORIC
    SEARCH IN SASTACK FOR A COMPATIBLE NP
    IF SUCCESS
      RESOLVE IT TO THAT NP
    ELSE{FAILED}
      SEARCH IN DASTACK FOR A COMPATIBLE NP
      IF SUCCESS
        RESOLVE IT TO THAT NP
        UPDATE DASTACK
      ELSE {FAILED}
        SEARCH IN BASTACK FOR A COMPATIBLE NP
        IF SUCCESS
          RESOLVE IT TO THAT NP
          UPDATE BASTACK, DASTACK
        ELSE {FAILED}
          FAIL RESOLUTION
        END IF
      END IF
    END IF
  END IF
END FOR

PUSH THE CURRENT SASTACK TO DASTACK
REMOVE DUPLICATES IN ALL STACKS
```

The anaphora resolution strategies adopted above can solve most of the cases cited in the example sentences. There are a few cases, however, that seem to require other approaches. For example, for (3c) the system must know that NPs having EACH attached to the head can be an antecedent of THEY. This is relatively easily done by carefully controlling the matching conditions. (3g) presents a case where pronouns are often used to refer to events, and a special mechanism must be constructed to deal with it.

Cases like (3i) and the instances cited in (4) (namely YOU, WE, THEY, and the first 3 cases of IT) must be treated as frozen constructions that require specialized conditions when the system is instructed to invoke the matching procedure. The last case of IT, cited in (4), is quite different from all others in that IT refers to the literal word, the linguistic entity, and as such it must also be treated independently. Lastly, the case of HE in (4) involves the system to infer the correct pragmatic relationship between John and Bill from the machine internal knowledge of some sort. The present

system is incapable of handling the cases where the complete understanding of anaphora calls for the availability of world-knowledge in the system.

IMPLEMENTATION DETAILS OF THE ASTACK

The precise configuration of the various kinds of ASTACK is given in the BNF grammar presented below.

```
(16) <sastack> ::= (<string><morph><sem><synt><nptype><intens>
                <ext>)*
<dastack> ::= (<sastack>)*
<bastack> ::= (<sastack>)*
<string> ::= (<ts>*)
<morph> ::= ([<num>])
<num> ::= sg | pl | any
<sem> ::= ([<human>][<animacy>][<gender>])
<human> ::= (human . <val1>)
<animacy> ::= (animate . <val1>)
<gender> ::= (gender . <val2>)
<synt> ::= ([<role>][<tier>])
<role> ::= (role . <val3>)
<tier> ::= (level . <int>)
<nptype> ::= (<val4>)
<val1> ::= t | nil
<val2> ::= m | f
<val3> ::= nom | do | io
<val4> ::= def | indef
<intens> is any LISP expression
<ext> is any LISP expression
<ts> is any linguistic object
<int> is any integer 0, 1, 2, ...
```

N.B. Brackets indicate optionality

A few items in (16) may need elaboration. The properties of animacy and humanness have a predictable relation that if an NP is marked (HUMAN . T), then it entails (ANIMATE . T). If any of these properties specified, all these properties are thought to have the value UNKNOWN.

The <synt> type of information may contain <role> and <domain>. <role> indicates what thematic role the NP is assigned and it is implemented to reflect the theta criterion in the GB Theory. As noted before the usefulness of this type of data is suggested by Hajicova & Sgall (1985). Not all NPs in the SASTACK are equally likely to be pointed to by the anaphoric expression in the next sentence, and, as they suggested, a ranking of topicalizability, perhaps of the form Subject NP > Direct Object NP > Indirect Object NP > Other Oblique NP will be useful in some cases.

<tier> contains information concerning the embedding level of the NP. A reordering of SASTACK elements according to the topicalizability will be done before the SASTACK is sent to DASTACK.

<ext> is any LISP expression and keeps track of the referents and/or the accessing function for the NP, and whatever other information needed for efficient computation. The field <ext> is designed to store the values of calls to access functions, or the access function to the extension of the NP, preventing a repeated execution of the same access function at a considerable savings of the CPU time.

The procedures relating to the ASTACK are verification actions on a pop arc in the parser. An alternative to this approach is to scan the complete parse tree for instances of anaphora, and then initiate necessary operations for resolution. This approach requires that the traditional (GB type) tree structure is available for inspection, but, unfortunately, the present parser maps the input sentence directly to a DBMS language (SQL) expression. The direct mapping approach is more computation-efficient since it does away with a parser-DBMS interface component.

In terms of anaphora resolution, the direct approach can foresee at least one complication in implementing the anaphora resolution algorithm in (15). As stated above, the anaphora resolution mechanism is invoked immediately before a pop arc is taken, where attempts are made to resolve anaphors using the information available at that point. It follows that any anaphora information appearing in the unseen string is not available for analysis. This means, then, any instance of anaphora which has its antecedent to its right will not be resolved (more precisely, the case of cataphora), such as the first three cases of IT in (4) and and THEY in (17).

(17) When they are angry, gorillas can be mean.

It is a little comforting to know that instances of cataphora are not common and probably even less common in database queries. An easy solution to this problem is to check for the presence of cataphor in the SQL parse, before it is executed. But this is effective only if the cataphor is resolved within the sentence, and the procedure fails completely if the resolution expands beyond the sentence boundary.

The last item that must be mentioned is the mechanism whereby full NPs are reduced to pronominal expressions. Although the majority of the cases will be handled by extracting the information necessary for pronominalization from the response to the query generated by the system and use it for correct pronominalization. A number of devices can be built in to emulate human responses. To list some heuristic ideas without further elaboration: a) if the answer is NIL or empty, use NONE, NO ONE, NOTHING, etc., b) if the number of elements retrieved is more than one, respond with THEY, c) if the query NP is of the kind that require the full NP repeated in the response (see (2g)), do so, etc. Much more work remains to be done in this area.

NOTES

- * An earlier version of this paper was presented at Linguistics Colloquy at the University of Kansas on 21 January 1986. My thanks go to the editors of KWPL and the referees who commented on the paper.
1. Michael Henderson correctly pointed out to me that the PERSON information is not as useful as the other types. since the first person always resolves to the user, and the second to the system. Since these two cases are easily recognizable. the rest can be assigned the third person marking by default.

REFERENCES

- Berwick, Robert C., and Amy S. Weinberg. 1984. The Grammatical Basis of Linguistic Performance: Language Use and Acquisition. Cambridge: MIT Press.
- Carpenter, P.A. and M. A. Just. 1977. Integrative Processes in Comprehension. Basic Processes in Reading. ed. by D. LaBerge and S. J. Samuels. Hillsdale: Earlbaum.
- Chang, F. 1980. Active Memory Processes in Sentence Comprehension: Clause Effect and Pronominal Reference. Memory and Cognition, 8:58-64.
- Chomsky, Noam. 1980. On Binding. Linguistic Inquiry, 11:1-46.
- Clark, Herbert. and C. J. Sengul. 1979. In Search of Referents for Nouns and Pronouns. Memory and Cognition, 7:35-41.
- Comrie, Bernard. 1981. Language Universals and Linguistic Typology. Chicago: University of Chicago Press.
- Geach, T. A. 1962. Reference and Generality. Ithaca: Cornell University Press.
- Grocz, B. J. 1977. The Representation and the Use of Focus in Dialogue Understanding. Technical Note 151, SRI.
- Grocz, B. J., A. Joshi, S. Weinstein. 1983. Providing a Unified Account of Definite Noun Phrases in Discourse. Technical Note 292, SRI.
- Guenthner, Franz. and Hubert Lehmann. 1984a. Automatic Construction of Discourse Structures. The Proceeding of the COLING84. pp. 398-401.
- , 1984b. Rules for Pronominalization. The Proc. of the First Conf. of the European Chapter of the ACL. pp. 144-151.

- Guindon, Raymonde. 1985. Anaphora Resolution: Short-Term Memory and Focusing. The Proc. of the 23rd Ann. Meeting of the ACL. pp. 218-227.
- Hajicova, Eva. and Petr Sgall. 1985. Towards an Automatic Identification of Topic and Focus. Proc. of the Second Conf. of the European Chapter of the ACL. pp. 263-267.
- Hobbs, J. R. 1976. Pronoun Resolution. Research Report 76-1. New York City College.
- Karttunen, Lauri. 1984. Features and Values. The Proceeding of the COLING84. pp. 28-33.
- , 1977. Syntax and Semantics of Questions. Linguistics and Philosophy. 1:3-44.
- Lasnik, H. 1976. Remarks on Co-reference. Linguistic Analysis, 2:1-22.
- Lesgold, A. M., S. F. Roth., and M. E. Curtis. 1979. The Foregrounding Effects in Discourse Comprehension. Journal of Verbal Learning and Verbal Behavior, 18:281-308.
- McKoon, G. and R. Ratcliff. 1980. The Comprehension Processes and Memory Structures Involved in Anaphoric References. Journal of Verbal Learning and Verbal Behavior. 19:668-682.
- Miller, G. A. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. Psychological Review, 63:81-97.
- Radford, Andrew. 1981. Transformational Syntax. London: Cambridge University Press.
- Reicheman, R. 1984. Extended Person-Machine Interface. Artificial Intelligence. 22:157-218.
- , 1978. Conversational Coherency. Cognitive Science, 2:283-327.
- Reilly, Ronan. 1984. A Connectionist Model of Some Aspects of Anaphor Resolution. The Proceeding of the COLING84. pp. 144-149.
- Sidner, C. 1983. Focusing in the Comprehension of Definite Anaphora. Computational Models of Discourse. M. Brady and R. C. Berwick, eds. Cambridge: MIT Press.
- , 1979. Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse. Technical Report 357. MIT Artificial Intelligence Laboratories.

- Stirling, Lesley. 1985. Distributives, Quantifiers and the Multiplicity of Events. Proc. of the Second Conf. of the European Chapter of ACL. pp. 16-24.
- Tennant, Harry. 1981. Natural Language Processing. New York: Petrocelli.
- Wasow, Thomas. 1979. Anaphora in Generative Grammar. Ghent: E. Story-Scientia P.V.B.A.