21

# INDIRECT RESPONSES TO LOADED QUESTIONS*

S. Jerrold Kaplan

Department of Computer and Information Science
University of Pensylvania
Philadelphia, Pa. 19104

Casual users of Natural Language (NL) computer systems are typically inexpert not only with regard to the technical details of the underlying programs, but often with regard to the structure and/or content of the domain of discourse. Consequently, NL systems must be designed to respond appropriately when they can detect a misconception on the part of the user. Several conventions exist in cooperative conversation that allow a speaker to indirectly encode their intentions and beliefs about the domain into their utterances, ("loading" the utterances), and allow (in fact, often require) a cooperative respondent to address those intentions and beliefs beyond a literal, direct response. To be effective, NL computer systems must do the same. The problem, then, is to provide practical computational tools which will determine both when an indirect response is required, and what that response should be, without requiring that large amounts of domain dependent world knowledge be encoded in special formalisms.

This paper will take the position that distinguishing language driven inferences from domain driven inferences provides a framework for a solution to this problem in the Data Base (DB) query domain. An implemented query system (CO-OP) is described that uses this distinction to provide cooperative responses to DB queries, using only a standard (CODASYL) DB and a lexicon as sources of world knowledge.

## WHAT IS A LOADED QUESTION?

loaded question is one that indicates that the questioner presumes something to be true about the domain of discourse that is actually false. Question 1A presumes 1B. A cooperative speaker must find 1B assumable (i.e. not believe it to be false) in order to appropriately utter 1A in a cooperative conversation, intend it literally, and expect a correct, direct response.

1A. What day does John go to his weekly piano lesson?
1B. John takes weekly piano lessons.
1C. Tuesday.

Similarly, 2A presumes 2B.

2A. How many Bloody Marys did Bill down at the banquet?
2B. Hard liquor was available at the banquet.
2C. Zero.

If the questioner believed 2B to be false, there would be no point in asking 2A - s/he would already know that the correct answer had to be "Zero." (2C).

Both examples 1 and 2 can be explained by a convention of conversational cooperation: that a questioner should leave the respondent a choice of direct answers. That is, from the questioner's viewpoint upon asking a question, more than one direct answer must be possible.

It follows, then, that if a question presupposes something about the domain of discourse, as 1A does, that a questioner cannot felicitously utter the question and believe the presupposition to be false. This is a result of the fact that each direct answer to a question entails the question's presuppositions. (More formally, if question Q presupposes proposition P, then each question-direct answer pair (Q, Ai) entails P*.) Therefore,

---

* This work partially supported by NSF grant MCS 76-19466

* This entailment condition is a necessary but not sufficient condition for presupposition. The concept of presupposition normally includes a condition that the negation of a

22

if a questioner believes a presupposition to be false, s/he leaves no options for a correct, direct response - violating the convention. Conversely, a respondent can infer in a cooperative conversation from the fact that a question has been asked, that the questioner finds it's presuppositions assumable. (In the terms of [Keenan 71], the logical presupposition is pragmatically presupposed.)

Surprisingly, a more general semantic relationship exists that still allows a respondent to infer a questioner's beliefs. Consider the situation where a proposition is entailed by all but one of a question's direct answers. (Such a proposition will be called a presumption of the question.) By a similar argument, it follows that if a questioner believes that proposition to be false, s/he can infer the direct, correct answer to the question - it is the answer that does not entail the proposition. Once again, to ask such a question leaves the respondent no choice of (potentially) correct answers, violating the conversational convention. More importantly, upon being asked such a question, the respondent can infer what the questioner presumes about the context.

Question 2A above presumes 2B, but does not presuppose it: 2B is not entailed by the direct answer 2C. Nonetheless, a questioner must find 2B assumable to felicitously ask 2A in a cooperative conversation - to do otherwise would violate the cooperative convention. Similarly, 3B below is a presumption but not a presupposition of 3A (it is not entailed by 3C).

3A. Did Sandy pass the prelims?
3B. Sandy took the prelims.
3C. No.

If a questioner believes in the falsehood of a presupposition of a question, the question is inappropriate because s/he must believe that no direct answer can be correct; similarly, if a questioner believes in the falsehood of a presumption, the question is inappropriate because the questioner must know the answer to the question - it is the direct answer that does not entail the presumption. In short, the failure of a presupposition renders a question infelicitous because it leaves no options for a direct response; the failure of a presumption renders a question infelicitous because it leaves at most one option for a direct response. (Note that the definition of presumption subsumes the definition of presupposition in this context.)

## CORRECTIVE INDIRECT RESPONSES

In a cooperative conversation, if a respondent detects that a questioner incorrectly presumes something about the domain of discourse, s/he is required to correct that misimpression. A failure to do so will implicitly confirm the questioner's presumption. Consequently, it is not always the case that a correct, direct answer is the most cooperative response. When an incorrect presumption is detected, it is more cooperative to correct the presumption than to give a direct response. Such a response can be called a Corrective Indirect Response. For example, imagine question 4A uttered in a cooperative conversation when the respondent knows that no departments sell knives.

4A. Which departments that sell knives also sell blade sharpeners?
4B. None.
4C. No deparments sell knives.

Although 4B is a direct, correct response in this context, it is less cooperative than 4C. This effect is explained by the fact that 4A presumes that some departments sell knives. To be cooperative, the respondent should correct the questioner's misimpression with an indirect response, informing the questioner that no departments sell knives (4C). (The direct, correct response 4B will reinforce the questioner's mistaken presumption in a cooperative conversation through it's *failure to state otherwise.)* A *failure* to produce corrective indirect responses is highly inappropriate in a cooperative conversation, and leads to "stonewalling" - the giving of very limited and precise responses that fail to address the larger goals and beliefs of the questioner.

## RELEVANCE TO DB QUERIES

Most NL computer systems stonewall, because their designs erroneously assume that simply producing the correct, direct response to a query insures a cooperative response. (To a great extent, this assumption results from the view that NL

---

proposition (in this case, the negation of the proposition expressed by a question-direct answer pair) should also entail its presuppositions. Consequently, the truth of a presupposition of a question is normally considered a prerequisite for an answer to be either true or false (for a more detailed discussion see [Keenan 73]). These subtleties of the concept of presupposition are irrelevant to this discussion, because false responses to questions are considered a-priori to be uncooperative.

functions in this domain simply as a high-level query language.) Unfortunately, the domain of most realistic DB's are sufficiently complex that the user of a NL query facility (most likely a naive user) will frequently make incorrect presumptions in his or her queries. A NL system that is only capable of a direct response will necessarily produce meaningless responses to failed presuppositions, and stonewall on failed presumptions. Consider the following hypothetical exchange with a typical NL query system:

Q: Which students got a grade of F in CIS500 in Spring, '77?
R: Nil. [the empty set]
Q: Did anyone fail CIS500 in Spring, 77?
R: No.
Q: How many people passed CIS500 in Spring, '77?
R: Zero.
Q: Was CIS500 given in Spring '77?
R: No.

A cooperative NL query system should be able to detect that the initial query in the dialog incorrectly presumed that CIS500 was offered in Spring, '77, and respond appropriately. This ability is essential to a NL system that will function in a practical environment, because the fact that NL is used in the interaction will imply to the users that the normal cooperative conventions followed in a human dialog will be observed by the machine. The CO-OP query system, described below, obeys a number of conversational conventions.

While the definition of presumption given above may be of interest from a linguistic standpoint, it leaves much to be desired as a computational theory. Although it provides a descriptive model of certain aspects of conversational behavior, it does not provide an adequate basis for computing the presumptions of a given question in a reasonable way. By limiting the domain of application to the area of data retrieval, it is possible to show that the linguistic structure of questions encodes considerable information about the presumptions that the questioner has made. This structure can be exploited to compute a significant class of presumptions and provide appropriate corrective indirect responses.

## LANGUAGE DRIVEN VS. DOMAIN DRIVEN INFERENCE

A long standing observation in AI research is that knowledge about the world - both procedural and declarative - is required in order to understand NL.* Consequently, a great deal of study has gone into determining just what type of knowledge is required, and how that knowledge is to be organized, accessed, and utilized. One practical difficulty with systems adopting this approach is that they require the encoding of large amounts of world knowledge to be properly tested, or even to function at all. It is not easy to determine if a particular failure of a system is due to an inadequacy in the formalism or simply an insufficient base of knowledge. Frequently, the collection and encoding of the appropriate knowledge is a painstaking and time consuming task, further hindering an effective evaluation. Most NL systems that follow this paradigm have a common property: they decompose the input into a suitable "meaning" representation, and rely on various deduction and/or reasoning mechanisms to provide the "intelligence" required to draw the necessary inferences. Inferences made in this way can be called domain** driven inferences, because they are motivated by the domain itself***.

While domain driven inferences are surely essential to an understanding of NL (and will be a required part of any comprehensive cognitive model of human intelligence), they alone are not sufficient to produce a reasonable understanding of NL. Consider the following story:

> John is pretty crazy, and sometimes does strange things. Yesterday he went to Sardi's for dinner. He sat down, examined the menu, ordered a steak, and got up and left.

For a NL system to infer that something unusual has happened in the story, it must distinguish the story from the events the story describes. A question answering system that would respond to "What did John eat?" with "A steak." cannot be said to understand the story. As a sequence of events, the passage contains nothing unusual - it simply omits details that can be filled in on the basis of common knowledge about restaurants. As a story,

---

* For example, to understand the statement "I bought a briefcase yesterday, and today the handle broke off." it is necessary to know that briefcases *typically* have handles.

** "Domain" here is meant to include general world knowledge, knowledge about the specific context, and inferencial rules of a general and/or specific nature about that knowledge.

*** Of course, these inferences are actually made on the basis of *descriptions* of the domain (the internal meaning representation) and not the domain itself. What is to be evaluated in such systems is the *sufficiency* of that description in representing the domain.

however, it raises expectations that the events do not. Drawing the inference "John didn't eat the steak he ordered." requires knowledge about the language in addition to knowledge about the domain. Inferences that require language related knowledge can be called language driven inferences.

Language driven inferences can be characterized as follows: they are based on the fact that a story, dialog, utterance, etc. is a description, and that the description itself may exhibit useful properties not associated with the thing being described.* These additional properties are used by speakers to encode essential information - a knowledge of language related conventions is required to understand NL.

Language driven inferences have several useful properties in a computational framework. First, being based on general knowledge about the language, they do not require a large infusion of knowledge to operate in differing domains. As a result, they are somewhat more amenable to encoding in computer systems (requiring less programming effort), and tend to be more transportable to new domains. Second, they do not appear to be as subject to runaway inferencing, i.e. the inferencing is driven (and hence controlled) by the phrasing of the input. Third, they can often achieve results approximating that of domain driven inference techniques with substantially less computational machinery and execution time.

As a simple example, consider the case of factive verbs. The sentence "John doesn't know that the Beatles broke up." carries the inference that the Beatles broke up. Treated as a domain driven inference, this result might typically be achieved as follows. The sentence could be parsed into a representation indicating John's lack of knowledge of the Beatles' breakup. Either immediately or at some suitable later time, a procedure might be invoked that encodes the knowledge "For someone to not know something, that something has to be the case." The inferencial procedures can then update the knowledge base accordingly. As a language driven inference, this inference can be regarded as a lexical property, i.e. that factive verbs presuppose their complements, and the complement immediately asserted, namely, that the Beatles broke up. (Note that this process cannot be reasonably said to "understand" the utterance, but achieves the same results.) Effectively, certain inference rules have been encoded directly into the lexical and syntactic structure of the language - facilitating the drawing of the inference without resorting to general reasoning processes.

Another (simpler) type of language driven inferences are those that relate specifically to the structure of the discourse, and not to it's meaning. Consider the interpretation of anaphoric references such as "former", "latter", "vice versa", "respectively", etc. These words exploit the linear nature of language to convey their meaning. To infer the appropriate referents, a NL system must retain a sufficient amount of the structure of the text to determine the relative positions of potential referents. If the system "digests" a text into a non-linear representation (a common procedure), it is likely to lose the information required for understanding.

The CO-OP system, described below, demonstrates that a language driven inference approach to computational systems can to a considerable extent produce appropriate NL behavior in practical domains without the overhead of a detailed and comprehensive world model. By limiting the domain of discourse to DB queries, the lexical and syntactic structure of the questions encodes sufficient information about the user's beliefs that a significant class of presumptions can be computed on a purely language driven basis.

## CO-OP: A COOPERATIVE QUERY SYSTEM

The design and a pilot implementation of a NL query system (CO-OP) that provides cooperative responses and operates with a standard (CODASYL) DB system has been completed. In addition to producing direct answers, CO-OP is capable of producing a variety of indirect responses, including corrective indirect responses. The design methodology of the system is based on two observations:

1) To a large extent, the inferencing required to detect the need for an indirect response and to select the appropriate one can be driven directly from the lexical and syntactic structure of the input question, and

2) the information already encoded in standard ways in DB systems complements the language related knowledge sufficiently to produce appropriate conversational behavior without the need for separate "world knowledge" or "domain specific knowledge" modules.

Consequently, the inferencing mechanisms required to produce the cooperative responses are domain transparent, in the

---

* In the story example, assumptions about the connectedness of the story and the uniformity of the level of description give rise to the inference that John didn't eat what he ordered. These assumptions are conventions in the language, and not properties of the situation being described.

sense that they will produce appropriate behavior without modification from any suitable DB system. These mechanisms can therefore be transported to new DB's without modification.

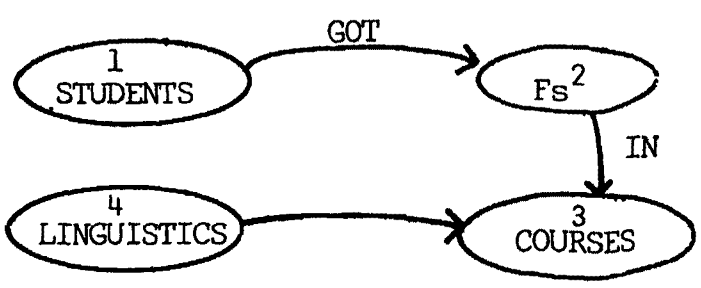
To illustrate this claim, a detailed description of the method by which corrective indirect responses are produced follows.

## THE META QUERY LANGUAGE

Most DB queries can be viewed as requesting the selection of a subset (the response set) from a presented set of entities (this analysis follows [Belnap 76]). Normally, the presented set is put through a series of restrictions, each of which produces a subset, until the response set is found. This view is formalized in the procedures that manipulate an intermediate representation of the query, called the Meta Query Language (MQL).

The MQL is a graph structure, where the nodes represent sets (in the mathematical, not the DB sense) "presented" by the user, and the edges represent binary relations defined on those sets, derived from the lexical and syntactic structure of the input query. Conceptually, the direct response to a query is an N-place relation realized by obtaining the referent of the sets in the DB, and composing them according to the binary relations. Each composition will have the effect of selecting a subset of the current sets. The subsets will contain the elements that survive (participate) in the relation. (Actually, the responses are realized in a much more efficient fashion - this is simply a convenient view.)

As an example, consider the query "Which students got Fs in Linguistics courses?" as diagrammed in FIGURE 1.




Meta Query Language representation of "Which students got Fs in Linguistics courses?"

FIGURE 1

This query would be parsed as presenting 4 sets: "students", "Fs", "Linguistics", and "courses". (The sets "Linguistics" and "Fs" may appear counterintuitive, but should be viewed as singleton entities assumed by the user to exist somewhere in the DB.) The direct answer to the query would be a 4 place relation consisting of a column of students, grades (all Fs), departments (all Linguistics), and courses. For convenience, the columns containing singleton sets (grades and departments) would be removed, and the remaining list of students and associated courses presented to the user.

Executing the query consists of passing the MQL representation of the query to an interpretive component that produces a query suitable for execution on a CODASYL DB using information associated for this purpose with the lexical items in the MQL. (The specific knowledge required to perform this translation is encoded purely at the lexical level: the only additional domain dependent knowledge required is access to the DB schema.)

The MQL, by encoding some of the syntactic relationships present in the NL query, can hardly be said to capture the meaning of the question: it is merely a convenient representation formalizing certain linguistic characteristics of the query. The procedures that mainipulate this representation to generate inferences are based on observations of a general nature regarding these syntactic relationships. Consequently, these inferences are language driven inferences.

## COMPUTING CORRECTIVE INDIRECT RESPONSES

The crucial observation required to produce a reasonable set of corrective indirect responses is that the MQL query presumes the non-emptiness of it's connected subgraphs. Each connected subgraph corresponds to a presumption the user has made about the domain of discourse. Consequently, should the initial query return a null response, the control structure can check the users presumptions by passing each connected subgraph to the interpretive component to check it's non-emptiness (notice that each *subgraph itself constitutes a well formed* query). Should a presumption prove false, an appropriate indirect response can be generated, rather than a meaningless or misleading direct response of "None."

For example, in the query of FIGURE 1, the subgraphs and their corresponding corrective indirect responses are (the numbers represent the sets the subgraphs consist of):

1) "I don't know of any students."
2) "I don't know of any Fs."
3) "I don't know of any courses."
4) "I don't know of any Linguistics."
1,2) "I don't know of any students that got Fs."
2,3) "I don't know of any Fs in

courses."
3,4) "I don't know of any Linguistics courses."
1,2,3) "I don't know of any students that got Fs in courses."
2,3,4) "I don't know of any Fs in linguistics courses."

Suppose that there are no linguistics courses in the DB. Rather than presenting the direct, correct answer of "None." the control structure will pass each connected subgraph in turn to be executed against the DB. It will discover that no linguistics courses exist in the DB, and so will respond with "I don't know of any linguistics courses." This corrective indirect response (and all responses generated through this method) will entail the direct answer, since they will entail the emptiness of the direct response set.

Several aspects of this procedure are worthy of note. First, although the selection of the response is dependent on knowledge of the domain (as encoded in a very general sense in the DB system - not as separate theorems, structures, or programs), the computation of the presumptions is totally independent of domain specific knowledge. Because these inferences are driven solely by the parser output (MQL representation), the procedures that determine the presumptions (by computing subgraphs) require no knowledge of the DB. Consequently, producing corrective indirect responses from another DB, or even another DB system, requires no changes to the inferencing procedures. Secondly, the mechanism for selecting the indirect response is identical to the procedure for executing a query. No additional computational machinery need be invoked to select the appropriate indirect response. Thirdly, the computational overhead involved in checking and correcting the users presumptions is not incurred unless it has been determined that an indirect response may be required. Should the query succeed initially, no penalty in execution time will be paid for the ability to produce the indirect responses. In addition, the only increase in space overhead is a small control program to produce the appropriate subgraphs (the linguistic generation of the indirect response is essentially free - it is a trivial addition to the paraphrase component already used in the parsing phase).

Corrective indirect responses, produced in this fashion, are language driven inferences, because they are derived directly from the structure of the query as represented by the MQL. If the query were phrased differently, a different set of presumptions would be computed. (This is not a drawback, as it might seem at first - it insures that the response will be in terms that the user understands.) For these reasons, corrective indirect responses, made possible by a careful choice of representations and associated algorithms, are produced in a domain transparent fashion with minimal system overhead using knowledge already available in the DB.

## A SHORT SAMPLE SESSION

This section contains an example interaction with the preliminary implementation of the CO-OP system as it now stands. An actual session has been abstracted and commented here.

CO-OP is currently implemented on a DEC KI-10 under TOPS-10 mainly in UCI-RUTGERS-LISP, and runs in 95K words, including the SEED DB system. SEED is a commercially available CODASYL DB system, marketed by International Data Base Systems, Inc. [Gerritsen 78]. The DB is a live one, used by the National Center For Atmospheric Research (NCAR) in Boulder Co. to keep track of their computer users, projects, research sponsors, etc. The primary purpose of the DB is to track the use of NCAR facilities, for accounting and other uses.

The user in the example is a hypothetical Computer Science department chairperson who has heard that the NCAR DB contains information about who is sponsoring research in various areas, and is interested in securing some research funds. S/he believes (incorrectly, it turns out) that funds are available to study mass storage. S/he begins by asking:

Q: Who sponsors active projects in mass storage?

R: I don't know of any projects in mass storage.

The system is unaware of any projects in mass storage, and so explicitly corrects the users presumption that there are some. The user tries a more general question.

Q: Who sponsors research in computer science?

R: I don't know of anyone that sponsors research in computer science.

Again, a corrective indirect response - but this time stating that no sponsors are known, rather than that no suitable projects are known. In addition to being able to generate corrective indirect responses, the system can choose the most appropriate one when more than one is applicable. This is done by observing that

27

the subgraphs of an MQL query are partially ordered according to a subgraph relation, and it is most appropriate to correct the "least" failing subgraphs with respect to this ordering.

Getting nowhere, the chairperson now asks:

Q: What are the areas?

and receives a list of the areas of interest (s/he knew that the projects at NCAR are grouped according to a predetermined list of areas of interest), identified by the area number and name. S/he observes that area #6 (data processing) and area #12 (numerical methods) look interesting, and follows up with:

Q: Who sponsors projects in area 6?

The response is a list of sponsor names with a supportive indirect component of the projects they sponsor in area 6, the name of the area (because only the number was supplied - the system doesn't currently remember that it just provided the area name to the user), and the project numbers of the sponsored projects. The user now decides that Nasa Headquarters looks the most promising, and so asks:

Q: What is sponsored in numerical methods by Nasa Headquarters?

After checking the DB, the system discovers that Nasa Headquarters doesn't sponsor anything in numerical methods. Additionally, it is unable to detect any failed presumptions on the part of the user. It therefore provides a negative response followed by a suggestive indirect response listing the projects that Nasa Headquarters sponsors in any area, in the hope that this will be helpful to the user.

R: I don't know of anything in numerical methods that Nasa Headquarters sponsors. But you might be interested in anything that Nasa Headquarters sponsors...

After perusing this list, the chairperson concludes that although the projects don't look very promising, s/he will get in touch with Nasa Headquarters. S/he asks:

Q: Who is the contact at Nasa Headquarters?

It turns out that there is a contact at Nasa Headquarters for each project sponsored, and so the system prints out the list (sorted by contact), along with the projects they sponsor. Although the user has presumed that there is only one contact at Nasa Headquarters, the system provides the entire list, without objecting. This and other forms of sloppy reference are tolerated by the system.

## CONCLUSION

The problem of producing apparently intelligent behavior from a NL system has traditionally been viewed in Artificial Intelligence as a problem of modelling human cognitive processes, or modelling knowledge about the real world. It has been demonstrated here that such approaches must include a pragmatic theory of the conventions and properties of the use of language, to function effectively. Domain driven inferences must be complemented by language driven inferences to appropriately process NL. Further, it has been argued that language driven inference mechanisms help to control the inference process, and can provide a more general and computationally attractive solutions to many problems previously thought to require domain driven inference.

A descriptive theory of one type of cooperative indirect response to inappropriate questions has been presented, and extended to a prescriptive (computational) theory by restricting the domain of application to DB query systems. This theory has been implemented using language driven mechanisms in the design of CO-OP, a cooperative query system. The result is the generation of appropriate corrective indirect responses in a computationally efficient and domain transparent fashion.

## REFERENCES


Austin, J.L., How To Do Things With Words, J.O. Urmson, Ed., Oxford University Press, N.Y. 1965.

Belnap, N. D., and T. B. Steel, The Logic of Questions and Answers, Yale University Press, New Haven, Conn., 1976.

Gerritsen, Rob, SEED Reference Manual, Version C00 - B04 draft, International Data Base Systems, Inc., Philadelphia, Pa., 19104, 1978.

Grice, H. P., "Logic and Conversation", in Syntax and Semantics: Speech Acts, Vol. 3, (P. Cole and J. L. Morgan, Ed.)

Academic Press, N.Y., 1975.

Harris, L. R., "Natural Language Data Base Query: Using the Data Base Itself as the Definition of World Knowledge and as an Extension of the Dictionary", Technical Report #TR 77-2, Mathematics Dept., Dartmouth College, Hanover, N.H., 1977.

Joshi, A. K., S. J. Kaplan, and R. M. Lee, "Approximate Responses from a Data Base Query System: An Application of Inferencing in Natural Language", in Proceedings of the 5th IJCAI, Vol. 1, 1977.

Kaplan, S. Jerrold, "Cooperative Responses from a Natural Language Data Base Query System: Preliminary Report", Technical Report, Dept. of Computer and Information Science, Moore School, University of Pennsylvania, Philadelphia, Pa., 1977.

Kaplan, S. J., and Joshi, A. K., "Cooperative Responses: An Application of Discourse Inference to Data Base Query Systems", to appear in proceedings of the Second Annual Conference of the Canadian Society for Computational Studies of Intelligence, Toronto, Ontario, July, 1978.

Joshi, A. K., Kaplan, S. J., and Sag, I. A., "Cooperative Responses: Why Query Systems Stonewall", to appear in proceedings of the 7th International Conference on Computational Linguistics, Bergen, Norway, August, 1978.

Keenan, E. L., "Two kinds of Presupposition in Natural Language", in <u>Studies in Linguistic Semantics</u>, (C. J. Fillmore and D. T. Langendoen, Ed.), Holt, Rinehart, and Winston, N.Y., 1971.

Keenan, E. L., and Hull, R. D., "The Logical Presuppositions of Questions and Answers", in <u>Prasuppositionen in Philosophie und Linguistik</u>, (Petofi and Frank, Ed.), Athenaum Verlag, Frankfurt, 1973.

Lee, Ronald M. "Informative Failure in Database Queries", Working Paper #77-11-05, Dept. of Decision Sciences, Wharton School, University of Pennsylvania, 1977.

Lehnert, W., "Human and Computational Question Answering", in <u>Cognitive Science</u>, Vol. 1, #1, 1977.

Searle, J. R., <u>Speech Acts, an Essay in the Philosophy of Language</u>, Cambridge University Press, London, 1969.

Weischedel, R. M., <u>Computation of a Unique Class of Inferences: Presupposition and Entailment</u>, Ph.D. dissertation, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa. 1975.

28