Computing Pronoun Antecedents in an English Query System

Kurt Godden
Computer Science Department
General Motors Research Laboratories
Warren, Michigan 48090

Abstract

In this paper I discuss how the Datalog English query system resolves pronominal references to extra-sentential antecedents that represent database records. When the system encounters a pronoun in a query, it searches through saved representations of earlier queries for an antecedent. A number of criteria must be satisfied before a proposed antecedent will be accepted. Among these are satisfaction of the pronoun's grammatical features and tests for contradictions and tautologies. Additional discriminators are applied in the event that there are two competing antecedents being considered. Of special interest is use of a hold queue mechanism which allows relaxation of the grammatical features of number and gender expressed by a personal pronoun. All of the strategies are independent of any application domain and do not fall into those parts of the system that need to be replaced or modified to interface Datalog to a new database.

1 Introduction

In this paper I will describe in detail the specific algorithms and heuristics of how pronominal antecedents are found in an English database front end, as currently implemented independently of the domain in the Datalog system [Godden, 1985, Hafner, 1984, Hafner and Godden, 1985]. Some of the system's abilities in responding to queries with pronouns are shown in (1). The examples of Datalog exchanges are taken from an implementation that interfaces to a simple fictional personnel database. The Datalog system including the pragmatics component is portable in the sense that it may easily be interfaced to other relational databases as well [Hafner and Godden, 1985]. In the examples, user input is in sans serif and Datalog's answers are in *slanted type.*

1 a. Does Smith work for the same dept as Stevens?
No, it's not true for the employee called Smith that department = electronics.
What dept does he work for?
Note: 'he' refers to a female.
Name *Department*
Smith *Physics*

b. Is the avg age of the men in electronics greater than 35?
Yes, for the 7 employees where sex is male and dept is electronics, the average age is 45.86
Is it more than the avg age of the women in electronics?
Yes, *it is true for those employees that average-age > 42*

c. How many men work in math?
There are 2 employees where sex is male and where department is mathematics.
Does Jones work for them?
No, it is not true for the employee called Jones that department is mathematics.

At present, Datalog supports three types of antecedents: *entities,* such as employees represented as database records (Ia); *data objects,* which are values that are computed such as the average age of the men (Ib); and *attribute values,* such as the name of an employee's department (Ic). This paper will discuss only the type of pronouns that refer to entities. For a discussion of the other two types see [Godden, 1988].

English has many different categories of pronouns as well as several linguistic devices for anaphoric reference. Datalog currently recognizes a limited subset of these, specifically the definite personal pronouns in their various inflected forms- *he, him, his, she, her, hers, it, its, they, them, their,* and *theirs.* The genitive forms (e.g. *his)* may be used as adjectival modifiers *(his age)* or may be nominalized *(greater than his).* As currently implemented, Datalog interprets these pronouns as having antecedents that appear in previous queries. The remainder of this paper discusses the methods used in this interpretation.

2 Knowledge Structures Used

Datalog uses frame-structured semantic representations which are constructed by the semantics component [Hafner, 1984]. The semantic frame structure that represents the pronoun *he* is shown in Figure 1.

```
(anaphor (var gl234) (q nil) (pro he)
 (restrictions
  (pref (att sex)(relop is)(value m))
  (pref(feat number)(relop is)(value sing))))
```

Figure 1. Semantic Frame for 'HE'

Of interest are the slots for *var* and *restrictions.* The purpose of the pragmatics component is to bind the variable in the *var* slot of an anaphor frame to its antecedent. The *restrictions* list contains the morphological features of the pronoun for number and gender which are used by pragmatics in testing potential antecedents. As explained below, these features are only *preferences* and not strict constraints on antecedents. The preference feature of number is always built for a pronoun frame. In contrast, the gender preference structure is only built when the underlying data can be distinguished by gender through an associated attribute such as *sex.* Further, the gender structure is only added to anaphor frames for singular pronouns, since plural pronouns are not so distinguished in English.

Potential antecedents are found in the discourse history that is maintained throughout a user session. Each parsed query is saved on this list and antecedent searching takes place in reverse order, inspecting the most recent query first. Each query is represented in the discourse history as a pair consisting of the semantic structure built for the query and a list of records retrieved in processing that query. If the list of records is found to represent the entities referred to by a pronoun, then the variable of that pronoun is bound to the list. A global parameter defines a horizon' beyond which the history list is not searched. This parameter has been arbitrarily set to allow consideration of the five preceding queries and has seemed adequate in use. If an antecedent is not found within the horizon, then the parse fails.

It was just mentioned that the structure pointed at by an anaphor's variable is a list of database records in the case of entity type antecedents. This list may come from a discourse history pair as noted above, or it may be a list that is the referent of a noun phrase substructure of some earlier query. All referring expressions, not just anaphors, have variables in their semantic structures that are bound to their referents. When it is determined that an anaphor refers to one of these referents, the anaphor's variable is set to the same list as that of the variable in the antecedent expression. Thus, entity-referring pronouns are interpreted as extensional expressions.

3 Control and the Hold Queue

This section presents the control algorithm used to search the discourse history list for an antecedent of a pragmatic pronoun. At the most general level, the algorithm first determines which of the three implemented types of antecedents to seek for a given pronoun and then it tries to find an antecedent of that type within the search horizon. In general, the desired antecedent type is easily determined from the overall semantic structure of the query containing the pronoun. For example as in Figure 2, if the subject of a query, called a topic in Datalog, is an anaphor and the predicate specifies some attribute-value restriction pair, then that anaphoric topic clearly refers to entities, or database records. This is because it is meaningless to predicate an attribute value of some computed data object, which is itself a value.

```
Do they work in math?
(query (perform (test true?))
 (topic(anaphor(var gl57)(q nil)(pro they)
       (restrictions
        (pref
(feat number)(relop is)(value plur)))))
 (predicate
  (prop (att dept)(relop is)(value math))))
```

Figure 2. Anaphor Referring to an Entity

On the other hand, the type of the intended antecedent cannot always be so easily determined. The query in Figure 3 could either be requesting a list of previously mentioned employees, or a list of ages or some other values just referred to by another query.

```
List them.
(query (perform (display))
 (topic(anaphor(var gl58)(q nil)(pro them)
       (restrictions
(pref (feat number)(relop is)(value plur)))))
 (predicate t))
```

Figure 3. Anaphor of Unknown Antecedent Type

In cases like this, the preceding query is inspected to determine what the referent is. It is assumed that if the immediately preceding query is a request for a data object, then the current anaphor is intended to refer to that value. Otherwise, the anaphor is taken to refer to the entities picked out by the preceding query. If the preceding query had no entities associated with it, as would be the case given a negative response to a yes/no question, then the entities chosen as antecedent are those referred to by the preceding query's topic.

Once the type of antecedent has been determined, control is transferred to the search algorithm, shown in simplified form in Figure 4. This top-level control algorithm searches back through the discourse history list checking each previous query's semantic structure for an antecedent of the appropriate type. The first appropriate antecedent found is returned.

```
1. Check next pair on history list for antec
2. If no antecedent found within horizon or
   if end of history list is reached,
      then if hold queue is not empty
              then print diagnostic and
                      return front of hold queue
              else return nil (parse fails)
      else if acceptable antecedent is found
              then return that antecedent
              else loop back to step 1.
```

Figure 4. Search Control Algorithm

The hold queue is used to store alternative antecedents. Alternative antecedents are those matching the desired type but differing from the pronoun in number or gender. How alternative antecedents are placed on the hold queue is discussed in section 4.1 below. The need for this hold queue became evident when targeted end users (and visiting researchers) made incorrect assumptions concerning the data which confused them during a session with Datalog. In (2), a typical example, the user assumes that Smith is a male by using *he.*

2 Tell me whether Smith works for the same dept as the tallest woman.
No, it is not true for the employee called Smith that department = computer-sci
What dept does he work for?

However, it turns out for the data in question that this assumption is incorrect because Smith is a woman. Without the hold queue, Datalog would either find no antecedent at all, or else a male antecedent from a different sentence leading to confusion since that was not the antecedent intended by the user. With the hold queue, however, the record for Smith is placed on the hold queue and the search for an antecedent continues. If no antecedent is found within the horizon that does satisfy the appropriate grammatical features as well as other tests to be described later, Smith's record is retrieved from the hold queue, and the system responds as shown in (3). A similar diagnostic is printed when the cardinality of the antecedent conflicts with the grammatical number of the pronoun.

3 *Note: 'he^9 refers to a female.*
<u>Name</u> Department
Smith Physics

As this example illustrates, the hold queue provides a fair degree of flexibility since the system prefers to satisfy the grammatical features of a pronoun when linking it to an antecedent, but these features may be relaxed.

4 Search Strategies for Entities

Let us now consider the procedure that searches a semantic query frame for an antecedent of type entity. In database queries, pronouns most likely refer to antecedents that correspond either to an entire query (4a) or to a semantic subject (4b).

4 Which men work in math?
a. Which of them (men in math) are over 40?
b. Which of them (men) work in physics?

Datalog first considers these potential antecedents before others referenced in predicates, embedded clauses, and other constituents. For convenience, let us call the entities referenced by an entire query as the *final set* and those referenced by a query's topic slot as the *topic set.* Recall that the final set of a query is saved on the history list paired with the semantic structure of the query that selected it.

4.1 Choosing between Topic Set and Final Set

There are four general strategies plus one default to determine which of the topic set or final set is the more appropriate antecedent. These strategies are invoked sequentially, the next being called only when the current strategy cannot make a choice. The final default strategy is used as a last recourse *after* attempting the other four. Even when a choice between topic set and final set is made, this choice is subject to placement on the hold queue.

The first strategy is for queries that result in an empty final set, as for a negative response to a yes/no question. For these cases, only the non-null topic set needs to be confirmed for reasonableness (cf. discussion of contradictions and tautologies below) to accept it as the antecedent. If both sets are null, they are not even considered since pronouns are assumed to refer successfully. When both the topic set and final set are non-null, the system cannot yet choose and moves to the next strategy. The other logical possibility is that the topic set is empty while the final set is not. This is an impossible case, however, since the final set is derived from the topic set and is, therefore, a subset of it.

A pair of specialized reasoning procedures are called to look for contradictions and tautologies as the second strategy. For example, if the assignment of one proposed antecedent would result in a contradictory or tautological reading involving the anaphor, then that proposed antecedent is rejected. This is how Datalog chooses the topic set as the antecedent in (5).

5 Are any of the men older than 30?
Are any of them younger than 30?

6 Is Bell in math?
What dept is Jones in?
Is he the same age as Jones?

Detection of contradictions and tautologies involves more than numeric comparisons, however. This is true in part because attributes may range over symbolic as well as numeric values. But even with numeric attributes comparisons are not always needed, as (6) illustrates. In (6), Jones is rejected as a possible antecedent without performing a numeric comparison on age and the system continues searching through earlier queries on the history list, where it finds and accepts Bell as the antecedent. It should be emphasized that the system computes these conditions symbolically and domain independently. If both the topic set and final set are ruled out due to resulting contradictions or tautologies, then the current query from the history list is examined for possible antecedents embedded elsewhere. If a potential antecedent is found in an embedded constituent, then it too is tested for contradiction or tautology and is subjected to a test for placement on the hold queue.

It is often the case that the topic set or final set cannot be chosen on the basis of contradictions or tautologies. The next strategy is a heuristic based upon the repetition of an attribute in the query with the pronoun. This heuristic is stated in (7).

7 The topic set is chosen as the antecedent if any attribute from the predicate of the proposed antecedent's query is repeated in the predicate of the query containing the anaphor.

As an example, consider the queries in (8). In (8a) the antecedent is taken to be the topic set: all men over 30. This contrasts with (8b) whose predicate does not contain a reference to the attribute *weight,* and here the system assigns the final set as the antecedent (see below).

8 a. How many men older than 30 weigh more than 200 pounds?
Which of them weigh less than 210 pounds?
b. How many men older than 30 weigh more than 200 pounds?
Which of them are taller than 70 inches?

The heuristic enforces the principle that a user's continued exploration of various ranges or values of an attribute expressed in the predicate is probably meant to discover different partitions based on that attribute of some fixed set of entities referred to by the anaphor. If the repeated attribute were instead intended to be a more specific narrowing of the previous set, then the first query itself would probably have contained a more specific restriction on the attribute in question. In (8a), if the user were really interested in seeing those men who are older than 30 and whose weight falls between 200 and 210 pounds, then he would have asked for that in his first query rather than using two queries. (However, see below.) This justification for the heuristic is well-motivated on general linguistic principles. The Least Effort Hypothesis [Zipf, 1949] of communication states that given alternative means to express some concept, language users will tend to choose that alternative that requires the least effort. While the Repeated Attribute Heuristic works in many instances, it is only an approximation and needs much improvement as (9) shows.

9 a. Are any employees heavy?
b. Are any of them tall?

Datalog assigns the final set of (9a) to the pronoun of (9b) as its antecedent since there is no repeated attribute and because the final set is the default assignment (see below). But the more natural reading of (9b) suggests that the topic set is the appropriate antecedent.

Notice also that although (9a) refers (via the adjective) to the attribute *weight* and (9b) to *height,* there is an important difference from the sequence in (8b). In (8b) there is a similar sequence of reference to the weight and height attributes, yet the assignment of the antecedent for (8b) to the final set seems correct in contrast to that same assignment in (9b). The difference seems to be that in the examples in (9) where fuzzy predicates (the adjectives *heavy* and *tall)* are used, there is the intuitive feeling that the attributes *weight* and *height* are somehow related in a way not exhibited by the use in (8) of specific reference to points in the domains of those attributes.

But there is another complicating factor involved here, namely the influence of syntactic and semantic parallelism. Sidner [1979] notes this same phenomenon and the difficulties it presents for her algorithm of anaphora resolution using focus. When we look at the sequence of (8a) a certain parallelism between the predicates is evident (where such parallelism is not so pronounced in (8b)). It is this same parallelism that may help explain why the favored interpretation of the pronoun in (9b) is to assign it to the topic set. If Datalog were to detect such parallelism (which it currently does not do), then this information could be used to avoid the default assignment of final set in (9b) noted above, and in similar examples where appropriate.

The fourth strategy compares the grammatical gender and number expressed by the pronoun against the same features found in the entities referred to by both the topic set and the final set. So for example, if the pronoun used is *she* the candidate antecedent sets are tested for the existence of a single female entity. Only the marked forms of *male* and *female* gender are tested. If the pronoun has neuter gender, e.g. *it,* no test is performed for gender. The queries in (10) show how Datalog responds to choices made by this strategy.

10 a. How many metallurgy employees *art* women?
There is 1 employee where department is metallurgy and where sex is female.
Is she older than 20?
Yes, it is true for that employee that age > 20.
b. How many metallurgy employees are women?
There is 1 employee where department is metallurgy and where sex is female.
Are they older than 20?
Yes, it is true for those employees that age>20

The resulting outcomes of comparisons using number and gender include the one where both potential antecedents match both features of the pronoun, whereupon no decision is made to choose one over the other. Another possibility is that neither of the potential antecedents matches either of the pronoun features. If this is the case, no decision is made and the potential antecedents are placed on the hold queue. A choice *is* made if only one feature matches only one of the proposed antecedents, or else if both features match the same proposed antecedent. In both situations, the decision of course favors the antecedent that is matched by the features.

If none of our preceding strategies was able to suggest a choice for antecedent, then the decision is arbitrarily made to choose the final set as antecedent. This subsumes the additional special case where the topic set is the same as the final set. Because this final decision is arbitrary, a message is printed to the user that indicates which antecedent is assumed by the system. An example of this is shown in (11), which precedes the answer to the second query of (8b).

11 *By 'them'1 assume you mean the employees where sex is male and age > 30 and where weight > 200.*

4.2 Pronoun Sequences as Implicit Focus

There is also a case where the antecedent is automatically chosen. This is to deal with a sequence of queries where the first specifies some referent that is referred to in succeeding queries by using pronouns. An example would be a sequence such as (12).

12 How many women are in math?
Do they have PhD's?
Which of them earn less than $40,000?

Once the first such pronoun in the sequence is bound to an antecedent (the women in math in this example), subsequent compatible pronouns are automatically bound to the same antecedent. For this purpose, compatible pronouns are defined to be those with the same number and gender (if applicable), but may have different case. Thus, *he* is compatible with *him* and *his,* but not with *it, she* or *they.* It should be pointed out that the sequence of queries with compatible pronouns need not be an unbroken sequence. There may be intervening queries with no pronouns at all, or some with pronouns that are not compatible with those of the sequence.

Treating sequences of compatible pronouns as coreferential can be viewed as a non-explicit use of the notion of focus. The first referential phrase selects what amounts to a focus—an object or set of objects which remains the antecedent of a subsequent series of pronouns. That original 'focus* will remain the antecedent of subsequent pronouns as long as those pronouns continue to be compatible with their predecessors *and* as long as the resulting bindings do not result in contradictory or tautological readings.

If an acceptable antecedent is not found using the methods described up to this point, then other structures are searched in the current query frame from the history list for entity-referring expressions. If any such expressions are found, then the proposed antecedent is tested for contradictions and tautologies, and is also subject to placement on the hold queue depending on how it matches the grammatical features of the anaphor. If no entity-referring expressions are found, then the search continues with the next query on the history list.

5 Problems

The first problem is that the current implementation requires that an antecedent occur as a *single* constituent. This means that in (13a) *they* is prevented from referring to both Jones and Smith.

13 a. Is Jones older than Smith?
Are they in the math dept?
b. Are Jones and Smith older than 35?
Are they in the math dept?

Therefore Datalog will skip over the first query in (13a) to inspect earlier queries in the discourse history list. This contrasts with (13b) where *they* does get properly bound to the combination of Jones and Smith, but the difference is that in (13b) Jones and Smith form a single constituent at both the syntactic and semantic levels. There is no straightforward way in which the discontinuous antecedents' of (13a) could be considered by the system. One approach that could be taken would be to build machinery to posit combinations of individually considered potential antecedents. This is the basic function of the *compose* operation of the NLC system [Ballard, 1982]. It remains to be seen in practice whether or not there is a great need for such operations in the environment of database queries.

A more general problem occurs when the antecedent the user had in mind is passed over by the system in favor of some other referent, or when the correct antecedent is not even considered because the system accepts another antecedent before the intended one is encountered.

This occurs most often as a result of a user's misconception concerning the number or gender of the intended antecedent. While the hold queue was designed for this eventuality, its use is sometimes circumvented due to the content of earlier queries within the search horizon. Consider again the example in (2), repeated here as (14), where *he* is intended to refer to Smith, who is a female.

14 Tell me whether Smith works in the same dept as the tallest woman?
What dept does he work for?

Another male referent may be chosen from an earlier query, before Smith can be retrieved from the hold queue. This could occur, for example, if Jones (a male) were referred to within the horizon of the search space in a context where all of the tests for inclusion or exclusion of an antecedent indicate Jones as an acceptable antecedent for the pronoun *he* in (14).

The sequence in (15) shows another situation where the wrong antecedent is chosen in the second query.

15 Which of the men is the shortest?
3 employees are *in this group.*

Name	Department	Height
Wilson	*Polymers*	*64*
Collier	*Polymers*	*64*
Bell	*Electronics*	*64*

Are they taller than the woman named Smith?
Yes, *it is true for 23 out of 26 of those employees that height > 66.*

Here, the user probably intends to refer to the three shortest men, but because of the repeated attribute heuristic the system binds the pronoun to the topic set: all 26 men in the database. One possible avenue out of this difficulty would be to use the notion of *perspective* as outlined by McCoy [1986]. In (15), the final set of the first query would be more salient than the topic set with regard to the attribute *height.* Since the second query asks about some group *(they)* from the perspective of height, the final set is therefore chosen. This would seem to be in direct conflict with the repeated attribute heuristic. Just how this conflict would be resolved remains for further study.

6 Relations to Other Work

Webber [1978] lists eight categories of objects that may serve as antecedents for anaphoric expressions. What I have referred to as entities coincides largely with her two categories of individuals and sets. The other two categories I mentioned, computed data objects and attribute values, are also fairly natural in the database domain but do not have neat counterparts in Webber's list. Webber takes the antecedent of an anaphor to be "the unique description of [a discourse entity] conveyed to the listener by the immediately preceding text" (p.28). In contrast, I have already pointed out how in Datalog pronouns refer directly to their antecedent objects in the world, i.e. the database. In [Godden, 1988] I discuss how Datalog's responses to queries with pronouns could be made more informative if the antecedent were an intensional structure instead. Such a change would put Datalog's treatment of the semantics of anaphors more closely aligned with Webber's view.

Some of the best-known work in discourse anaphora involves the notion of focus [Grosz, 1977, Sidner, 1979]. However, the concept of focus, while appearing to be especially well-suited to task-oriented dialogues, seems less well-suited to free-wheeling database dialogues where there is no specific task or goal to naturally constrain the flow of discourse. Therefore, the intent of the current investigation has been to explore alternative strategies for discourse anaphora while retaining the attempt to base these strategies on sound linguistic principles.

Brennan, Friedman, and Pollard [1987] discuss an algorithm for pronoun resolution based on the notion of centering [Grosz *et al.,* 1983]. While they state that their algorithm has been implemented in a natural language interface to a database query system, all their examples are taken from short story-like narratives. They mention that their system shares some similarities with Sidner's notions of focusing, but without database examples it is difficult to evaluate the appropriateness of their approach vis-a-vis my criticism of focusing for this application area. Their system does account for agreement features between a pronoun and its antecedent and also deals with coreference constraints, which Datalog does not. On the other hand, they reserve for future research use of information present such as was discussed previously under contradictions, tautologies, and the heuristics used by the Datalog system. In their system, potential antecedents (the "forward looking centers") are rank ordered by their participation in grammatical relations. As a result, the system favors as an antecedent the "subject, object, and object2, followed by other subcategorized functions, and finally adjuncts." (p. 156) Datalog favors such potential antecedent structures in a similar order, although it does so indirectly due to the search order for possible antecedents in the history list.

Rich and LuperFoy [1988] discuss a blackboard architecture used in the Lucy system for anaphora resolution in a front end for knowledge-based systems. Lucy handles both bound and pragmatic pronouns. There is a separate module for each strategy used. Essentially, the different modules specify potential antecedents along with a score and a confidence factor for that score. Another procedure selects the "best" of the possible antecedents after this scoring is done by each strategy module. In contrast with Datalog, if the system cannot make a choice, Lucy asks the end-user to choose among the alternative antecedents. Lucy employs some strategies that have no counterpart in Datalog, and vice versa. Some strategies in Lucy deal with bound pronouns (e.g. the familiar structural constraints on coreference) and others are applicable to pronouns in general, such as "semantic consistency" (selectional restrictions). Where the strategies used by Datalog and Lucy overlap, there are significant differences in their use, e.g. Lucy strictly enforces the constraints of number and gender, and recency is found as an overt strategy module in Lucy, as is focusing.

7 Conclusions

In this paper I have discussed how the Datalog system resolves pronominal references to extra-sentential antecedents. Of special interest are a) the five strategies used to decide between competing potential antecedents of type entity, and b) use of the hold queue mechanism to relax the grammatical features of number and gender expressed by a personal pronoun. All of the strategies used by the system are independent of any application domain and do not fall into those parts of the system that need to be replaced or modified to interface Datalog to a new database.

References

[Ballard, 1982] B ruce W. Ballard. Pronominal processing in a computational domain. Research Report CS-1982-1, Durham, NC: Duke University, 1982.

[Brennan *et al.,* 1987] Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics,* pp. 155-162, Stanford, CA: ACL, June 1987.

[Godden, 1985] Kurt Godden. Categorizing natural language queries for intelligent responses. In *Proceedings of the 1985 National Computer Conference,* pp. 67-73, Reston, VA: AFIPS Press, July 1985.

[Godden, 1988] Kurt Godden. Strategies for interpreting pragmatic pronouns. Research Publication GMR-6208, Warren, MI: General Motors Research Laboratories, 1988.

[Grosz, 1977] Barbara J. Grosz. *The representation and use of focus in dialogue understanding.* Technical Note 151, Menlo Park, CA: SRI International, 1977.

[Grosz *et al.,* 1983] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics,* pp. 44-50, Cambridge, MA: ACL, June 1983.

[Hafner, 1984] Carole D. Hafner. Interaction of knowledge sources in a portable natural language interface. In *Proceedings of the 10th International Conference on Computational Linguistics,* pp. 57-60, Stanford, CA: ACL, July 1984.

[Hafner and Godden, 1985] Carole D. Hafner and Kurt Godden. Portability of Syntax and Semantics in Datalog. In *ACM Transactions on Office Information Systems,* 3(2):141-164, April 1985.

[McCoy, 1986] Kathleen F. McCoy. The ROMPER System: Responding to Object-Related Misconceptions using Perspective. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics,* pp. 97-105, New York, NY: ACL, June 1986.

[Rich and LuperFoy, 1988] El aine Rich and Susann LuperFoy. An Architecture for Anaphora Resolution. In *Proceedings of the Second Conference on Applied Natural Language Understanding,* pp. 18-24, Austin, TX: ACL, 1988.

[Sidner, 1979] Candace L. Sidner. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse.* Technical Report 537, Cambridge, MA: MIT AI Lab, 1979.

[Webber, 1978] Bonnie Lynn Webber. *A Formal Approach to Discourse Anaphora.* Report No. 3761. Cambridge, MA: Bolt, Beranek and Newman, Inc., 1978.

[Zipf, 1949] George K. Zipf. *Human Behavior and the Principle of Least Effort.* Cambridge, MA: Addison-Wesley Press, Inc., 1949.