

NATURAL LANGUAGE QUERYING OF HISTORICAL DATABASES

James Clifford

Information Systems Area
Graduate School of Business Administration
New York University
New York, NY

In this paper we examine the connection between two areas of semantics, namely the semantics of historical databases and the semantics of natural language querying, and link them together via a common view of the semantics of time. Since the target application domain is an historical database, we present the essential features of the Historical Relational Database Model (HRDM), an extension to the relational model motivated by the desire to incorporate more "real world" semantics into a database at the conceptual level. We then present the essential features of QE-III, a formally defined English database query language whose semantic and pragmatic theory, based on a Montague-type semantics, makes explicit reference to the notion of denotation with respect to a moment of time. We demonstrate the use of this language to query an example historical database, and discuss the issues of how to provide both a semantic and a pragmatic interpretation for questions within a model-theoretic framework.

1 INTRODUCTION

The **relational model of data** (RM), first proposed in 1970 (Codd 1970), has by now become the standard for both database practitioners and theoreticians alike. In spite of this success, however, much recent database research has focused on ways to extend RM to overcome perceived shortcomings. Chief among the criticisms has been RM's lack of any "real-world semantics." Among the many diverse efforts directed at this deficiency have been a number of attempts to extend RM to incorporate a temporal dimension at the model level. While such efforts as Ben-Zvi (1982), Ariav et al. (1984), Snodgrass (1984), Lum et al. (1984), Clifford (1985), Snodgrass and Ahn (1985), and Gadia and Vaishnav (1985) have all addressed this issue, the **Historical Relational Database Model** (HRDM) ((Clifford 1982a), (Clifford and Warren 1983), (Clifford 1985), (Clifford and Croker 1987)) has the advantage of being directly parallel to a formal theory of natural language. In Section 2 we present an overview of HRDM, as it serves as the environment in which we wish to explore our query language. In particular, HRDM views database attributes as functions from moments in time to values (in the appropriate domain), and the intensional logic IL_s provides a mechanism for direct reference to these higher-order objects, and for incorporating them into a general temporal semantics for the database. We can therefore express

both static and dynamic queries in the same language, by quantifying over variables of the appropriate types.

In a series of papers culminating in Montague (1973), henceforth PTQ, Richard Montague embarked upon a program of providing a formal syntax coupled with a model-theoretic semantics for increasingly sophisticated fragments of English. Section 3 argues that a successful formal treatment can be given to a natural language querying facility for a **historical relational data base** (HRDB), through the medium of the intensional logic IL_s .

We view this work as important for two reasons. First, it represents one of the initial attempts to adapt the ideas of **Montague Semantics** (MS) (Montague 1974) to a practical problem. (Landsbergen (1981) looks at the issue of machine translation within an MS framework.) The research that has been done since the PTQ paper has primarily looked at extensions or modifications to its linguistic or logical theory, or at implementations of the theory on the computer. We will attempt to show that this theory of language can serve as the formal foundation of a useable computer system for querying actual databases.

Second, in addition to approaching the problem of NLQ formally, rather than from a purely engineering approach, the theory presented provides a novel (but see Gunji (1981) for a similar approach developed

Copyright 1988 by the Association for Computational Linguistics. Permission to copy without fee all or part of this material is granted provided that the copies are not made for direct commercial advantage and the CL reference and this copyright notice are included on the first page. To copy otherwise, or to republish, requires a fee and/or specific permission.

0362-613X/88/010010-34\$03.00

EMP	<u>EMP-NAME</u>	MGR	DEPT	SAL	
	John	John	Linen	25K	
	Mike	John	Linen	17K	
	Elsie	Elsie	Toy	26K	relation emp
	Liz	Liz	Hardware	30K	
	Rachel	Liz	Hardware	29K	
	Peter	Liz	Hardware	29K	

Figure 2-1. Example Relation.

concurrently with ours) approach to the interpretation of queries that involves both a semantic and a pragmatic account. This work represents only a first step in this direction within a MS framework. The fragment of English that we define herein is certainly not adequate to express all of the queries that one would want to present to an HRDB. It is intended only to lay the groundwork for a formal theory of database querying that is both extendible and implementable.

In this paper we present an informal overview of a fragment of English for database querying that we call **QE-III**. We discuss the kinds of properties and abilities that a database query language in English should possess; principal among these are (1) an account of question semantics that possesses close analogs in database theory, (2) an account of the semantics of multiple-WH questions, (3) an account of the semantics of time, and (4) a grammar that is conducive to a computer implementation. After examining a number of partial solutions to these problems, we introduce the notion of a formalized pragmatics as an equal partner with the syntax and semantics in the specification of the QE-III language. We argue that assigning to the pragmatic component the task of providing a representation for the answer(s) to a question is both appropriate and elegant. Finally, we discuss several other recent attempts at developing a formal theory of questions.

QE-III is defined as a formal language, with syntax paired with semantics, and with a pragmatics defined on the two of these; the language as a whole is designed with the database application in mind. QE-III is both a simplification and an extension of the PTQ semantic theory. Within the tradition of Montague Semantics, QE-III is a formalized fragment of English allowing questions, tenses, and temporal operators. The inclusion of a formal pragmatics as an interpretive component of QE-III is an interesting extension to the traditional conception of a Montague Grammar. Among the other extensions to the PTQ fragment embodied in QE-III are (1) the inclusion of time-denoting expressions and temporal operators, (2) an analysis of verb meanings into primitive meaning units derived from the database schema, and, of course, (3) the inclusion of certain forms of direct questions. These extensions, and the semantic and pragmatic interpretations with which

they are provided, are motivated by the ultimate goal of database access, but they are equally interesting in their own right. The syntactic theory presented is in some cases admittedly naive, for we have been primarily interested in getting the interpretation right.

Section 4 provides an overview of the salient features of the QE-III by means of a number of example derivations and translations. The complete definition of QE-III is given in Clifford (1982b) and again in Clifford (1987), where it appears with a fuller set of examples. We conclude in Section 5 with a discussion of some of the limitations of the fragment and of some possibilities for further extensions.

2 THE HISTORICAL RELATIONAL DATABASE MODEL

Analogous to the relationship between the relational model of data and first-order logic (Gallaire and Minker 1978), we can view an HRDB as a model for IL_s (Clifford 1982b). The higher-order language IL_s (with its built-in concept of denotation with respect to an index) provides a formal semantics for such data bases in a natural way.

In the standard or "static" relational model, we might see a relation such as *emp* on a scheme $EMP(EMP-NAME MGR SAL DEPT)$. A typical query of such a relation, say, "What is employee John's salary?", would be expressed in the relational algebra as $\pi_{SAL}(\sigma_{EMP-NAME=John}(emp))$. A first-order language would express this same query as something like $\{z \mid \exists x \exists y emp(John, x, y, z)\}$ where x, y , and z are individual variables and John is an individual constant. To answer such a query, a **data manipulation language (DML)** would access the current relation instance *emp* on *EMP*, such as the one in Figure 2-1.

More complex queries about the employees in this company, such as: (1) "Has John's salary risen?", (2) "When was Peter rehired?", (3) "Did Rachel work for the toy department last year?", (4) "Has John ever earned the same as Peter?", or (5) "Will the average salary in the linen department surpass 30K within the next 5 years, if current trends continue?" have typically not been expressible in any query language, because neither the structures nor the operations in the underlying data model provide for them.

In practice, database administrators have had to resort to ad-hoc solutions, typically involving programming in some host language, in order to handle queries of this sort. The issue of modeling time in a DBMS has recently attracted considerable attention within the database community. The HRDM (and other historical or temporal database models) attempts to satisfy the need for access to the temporal dimension of information by providing a unified and formal theory of database semantics that includes time. In particular, HRDM and QE-III, recognizing the need for maintaining a historical record of changing data, and a language (English) that makes explicit or implicit reference to the concept of time, together provide a theory of database semantics capable of interpreting sentences in the language correctly, i.e., in a way that corresponds with our intuitive understanding of the relation of time to the semantics of the real world.

Consider again the query "Has John's salary risen?" Even with time represented explicitly in the database, there is no apparent simple relational algebraic formulation for this query. With the first-order representation for John's salary given above, as a first guess we might imagine that $\text{RISE}(\{z \mid \exists x \exists y \text{ emp}(\text{John}, x, y, z)\})$ would represent this new query, where RISE is a predicate symbol. However, even with the knowledge that John has only one salary, say 25K, it clearly makes no sense to ask whether 25K "rises." To answer this question, more data is needed than the current extension of John's salary: the values of John's salary for some other point(s) of time (in this specific instance, in the past) are needed. The HRDM model presented in Clifford and Croker (1987), built upon a formalization of the concept of intension, provides a uniform way to view attributes (such as SALARIES) not as individual dollar amounts, but as functions from moments in time to dollar amounts. For the purposes of this paper we will present an overview of HRDM and discuss some of the issues involved.

Informally speaking, tuples in a relation represent facts about some object (entity or relationship) identified by the value of the key attribute(s). For example, in relation emp on scheme EMP (EMP-NAME MGR DEPT SALARY), the attribute EMP-NAME is the key attribute, and DEPT, MGR, and SALARY define properties of employees. A particular tuple, e.g., <Peter, Hardware, Maria 30K> represents facts about the employee Peter. A relation in the ordinary, or static, relational data model would consist of a set of such tuples representing the facts about a set of employees. Each tuple would consist of exactly four atomic values, one for each of the four attributes in the scheme.

By contrast, in HRDM a relation would provide historical information about the changing values of the attributes of the objects denoted by values of the key, in this instance about EMPLOYEES. Each tuple would be a complex, three-dimensional object whose size would be based upon what we call the lifespan of that particular

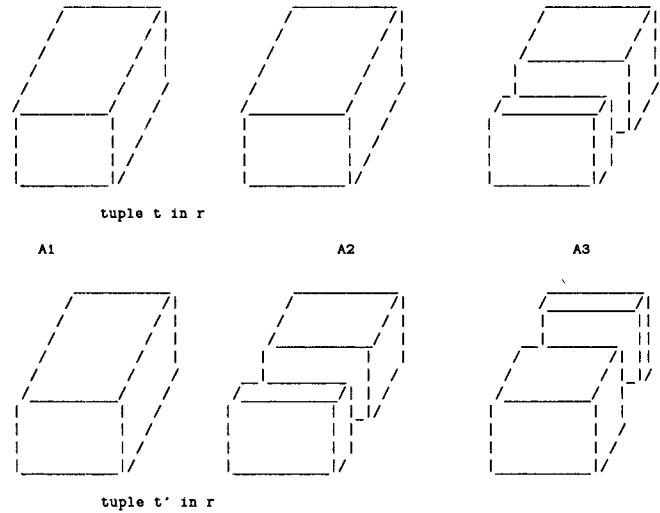


Figure 2-2. The History of Two Similar Objects.

employee, i.e., the times when that employee was of interest to the enterprise. Figure 2-2 depicts two tuples in the same relation but with different lifespans.

Time is represented in the HRDM as a set $T = \{ \dots, t_0, t_1, \dots \}$, at most countably infinite, over which is defined the linear (total) order $<_T$, where $t_i <_T t_j$ means t_i occurs before (is earlier than) t_j . (For the sake of clarity we will assume that $t_i <_T t_j$ if and only if $i < j$.) The set T is used as the basis for incorporating the temporal dimension into the model. We assume that T is isomorphic to the natural numbers, and therefore the issue of whether to represent time as intervals or as points is simply a matter of convenience. Using the natural numbers allows us to restrict our attention to closed intervals (a closed interval of T , written $[t_1, t_2]$ is simply the set $\{t_i \mid t_1 \leq t_i \leq t_2\}$).

$D = \{D_1, D_2, \dots, D_{n_a}\}$ is the set of value domains, where for each i , $D_i \neq \phi$. Each value domain D_i is analogous to the traditional database notion of a domain in that it is a set of atomic (non-decomposable) values. In HRDM, however, attributes take their values not from these simple domains, but rather from more complex functions. $U = \{A_1, A_2, \dots, A_{n_a}\}$ is a (universal) set of attributes. Simplifying somewhat, we define over the sets T and D a set of temporal mappings from the set T into the set D . This set, $TD = \{TD_1, TD_2, \dots, TD_{n_a}\}$ where for each i , $TD_i = \{f_i \mid f_i : T \rightarrow D_i\}$, is the set of all partial functions from T into the value domain D_i .

The domain of each attributes in HRDM is some set of partial temporal functions. Since key attributes are intended to be time-invariant, they are constrained to take a constant valued function (i.e., one which associates the same value with every time in its domain) as their value. As we shall see, these mappings are the counterparts to the notion of individual concepts in the intensional logic IL_s .

The notion of a tuple t on scheme R is expanded in HRDM to be an ordered pair, $t = \langle v, l \rangle$, where

1. $t.l$, the lifespan of tuple t , is a subset of the set T , and represents the set of times over which its attributes are defined, and
2. $t.v$, the value of the tuple, is a mapping such that \forall attributes $A \in R$, $t.v(A)$ is a mapping in $t.l \rightarrow \text{DOM}(A)$ (the value-domain of attribute A).

2.1 EXAMPLE DATABASE

In the remainder of the paper we will discuss the semantics and pragmatic theory of QE-III, illustrated with example database queries to an HRDB. For this purpose we now define the relation schemas for a historical department store data base based upon an example in Chang (1978).

EMP_REL (EMP MGR DEPT SAL)
 DEPT_REL (DEPT FLOOR)
 ITEM_REL (ITEM TYPE)
 SALES_REL (DEPT ITEM VOL)

This concludes our brief overview of HRDM. For further details the reader is referred to Clifford (1985) and Clifford and Croker (1987).

3 OVERVIEW OF ENGLISH QUERY LANGUAGE QE-III

3.1 INTRODUCTION

HRDM serves to formally incorporate a temporal semantics into an extended relational database model. In order to query a historical database using English, we define the semantics of queries expressed in English in terms of the semantics of HRDM, by defining a small query fragment as a Montague Grammar. The correlation between the database semantics and this query language is made explicit by providing the semantics of the query fragment via an indirect translation into the intensional logic IL_s .¹ The translations provide for a completely extensional treatment of verbs (i.e., there are no verbs like *seek*, which can be nonextensional in object position in the PTQ treatment). This treatment is dictated by the application to a database environment, in which existence is tantamount to existence in the database (Reiter 1978). Through these translations, then, the historical database essentially serves as a model for IL_s and therefore as the model for a formal definition of the interpretation of the English queries. In addition to providing a semantic interpretation, which in model-theoretic terms is called its **denotation**, we also provide for each expression a **pragmatic interpretation** in a manner to be explained.

Our goal in this effort has not been to define an English database query language that is, in any sense of the term, complete. Rather, we have been motivated by two complementary goals. First, we have wanted to investigate the possibility of a formal, model-theoretic query language for historical databases. This led to our interest in Montague Semantics and to our second goal—demonstrating that Montague's theories of natural language semantics are applicable to such a practical

task. Along the way we discovered that it was simpler and more natural to define the interpretation of this query language in two components, one semantic and the other pragmatic.

Two overriding principles have guided this work. First was that whatever interpretation or meaning our theory would give to a natural language database query should be as close as possible to the interpretation given to database queries in, say, the relational algebra or calculus. This meant that the interpretation of a query should somehow encompass its answer as represented in the underlying data base. Second, the theory should make sense computationally. This meant taking into account what had already been learned about parsing strategies for Montague Grammars (Friedman and Warren (1978), Warren (1979), Landsbergen (1981)), as well as what database theory had to say about the semantics of the modeled enterprise. These principles motivate certain systematic simplifications to the PTQ translations from English to logic, wherever these are suggested by the simplified view of the semantics of the enterprise provided by the database model. Moreover, since we are not attempting to develop a semantic theory of questions for English in general, these simplifications have been introduced into the translation process as early as possible. We believe that this strategy has the dual effect of making some of the PTQ theory a little more accessible, and eliminating the need to resort to the less computationally attractive technique of introducing a large number of meaning postulates and using logical equivalences to perform the reductions at a later stage.

We have made little attempt to develop a sophisticated syntax for our fragment. Numerous extensions to the syntax of the PTQ fragment have been investigated by researchers in the past decade that we have not incorporated into our fragment. Since our primary concern has been "getting the meaning right," we felt that a too broad syntactic coverage might obscure our major points. For this reason, we have extended the PTQ fragment only slightly. The treatment of questions that we present is syntactically naive, although in its favor we might point out that unlike most work on questions in Montague Grammar, QE-III makes a stab at direct questions. We believe that the semantic theory of questions that we present, and particularly our proposal to capture the answer in a pragmatic component, are an important contribution to the formalization of the interpretive component of natural language understanding systems. Naturally, the true test of a natural language query facility is in how useable it is; certainly the syntax of QE-III would have to be extended before anyone would think of using it.

In this section we discuss the major issues underlying the definition of QE-III, which fall roughly into two broad categories: aspects of the process of database querying that we have incorporated into the fragment, and modifications and additions to the PTQ fragment

that these, and the database semantics, have occasioned. As in much of the work that has been done in the area of Montague Semantics since Montague's death in 1970, we have allowed the PTQ fragment to stand pretty much intact as the heart of QE-III. However, we have redefined this fragment in terms of the language IL_s , in order to allow direct reference to moments in time.

3.2 PRELIMINARIES

3.2.1 INDIVIDUAL CONCEPTS VS. ENTITIES

Most recent research in the field of Montague Semantics has incorporated the suggestion, first made in Bennett (1974), that Montague's treatment of common nouns (CNs) and intransitive verbs (IVs) as denoting sets of individual concepts (ICs) is unduly complicated. Under Bennett's suggestion, both CNs and IVs denote sets of simple individuals, with the result that the entire typing scheme of the English categories in these fragments is considerably simplified. In Section 2 we showed that attributes in an HRDB can be identified with ICs. Accordingly we have not adopted the Bennett type system, but have instead maintained the treatment of PTQ.

3.2.2 VERBS

Montague's semantic treatment of verbs leaves them completely unanalyzed; thus, for example, the English verb "walk" translates into the constant "walk'" in IL , "love" into "love'", etc. The interpretation of these constants is some function in the model for the language, a function about which Montague says nothing except to specify its logical type (and in certain cases to specify an extensional meaning postulate). Because we are using a database as a representation of the logical model we are in a position to provide an analysis of English verbs that takes into account the meaning of the verbs as encoded in the database. This analysis is given in terms of the database schema. For example, instead of translating the verb "manage" into the unanalyzed predicate "manage'", we take advantage of the database semantics to incorporate directly into its translation the information that its subject must be an IC in the role of a MGR, and that its object must be a constant IC that is an EMP. We do not change the logical type of the translation, i.e., a transitive verb in our fragment denotes the same kind of function as it does in Montague's treatment; we simply analyze its meaning in terms of the database primitives. This analysis in terms of a small set of primitive meaning units is not very different from some approaches taken in AI work in natural language understanding (e.g., Schank 1972), or from the linguistic theory of deep cases (Fillmore 1968). The difference, of course, is that our primitives or cases are different, motivated by the HRDM and the schema design, and are no more absolute than any well-chosen database design.

As an example, the translation of "manage" in our

fragment is given as: $\lambda W\lambda xW(i)(\lambda yAS-1(y(i),x) \wedge EMP_*(i)(y(i)) \wedge MGR'(i)(x))$. This expression is of the same logical type as 'manage' in a PTQ-like treatment, and will combine with terms in the same way, but it does not leave "managing" unanalyzed. Instead it specifies what attribute class(es) its subject and object must belong to, and how they must be related. Specifically, the subject must be an entity ($y(i)$) that is an EMP, the object an IC (x) that is a MGR, and the MGR-IC must be ASSOCIATED with the EMPLOYEES (AS-1). In general the translation of any verb in our theory will so specify the attribute of its subject (or the disjunction of alternatives, if any). The translation of a TV will further specify the attribute(s) of its direct object, and of a DTV of both its objects. Moreover any relationship(s) among these attributes will also be specified.

3.3 THE PROBLEMS OF TENSE AND TIME

3.3.1 INTERVALS OR STATES?

David Dowty (1979) presents a discussion of a broad spectrum of semantic and syntactic issues relevant to the understanding of English, and in particular to providing a Montague Semantic analysis of these issues. In the final chapter of this book he formalizes many of the ideas he has discussed by defining a Montague fragment of English that includes such features as temporal adverbs, dative-taking verbs, a theory of word formation, and a treatment of several compound tense structures. In order to provide a semantics for this expansion of the PTQ fragment Dowty argues for the necessity of several significant extensions to the logic IL : a radically different treatment of the phenomenon of tense is one of his contributions. Because we are concerned with many of the same issues as Dowty—in particular tenses and direct temporal references—it seems appropriate to discuss his work and to contrast two different solutions to some of the same issues.

A major section of the book is concerned with developing a rigorous taxonomy of verbs in English based upon several syntactic and semantic criteria. The problems with a number of different classification schemes that have been proposed over the years are discussed, in particular Vendler's scheme (Vendler 1967), which divides verbs into the four categories of statives, activities, accomplishments, and achievements. Dowty judges all of these proposals by the two criteria of syntactic and semantic uniformity: can all of the verbs assigned to a given class appear in the same syntactic constructs, and are the same inferences in meaning justified for all like-classified verbs? Dowty's final taxonomy, offered with many reservations, defines eight different verb categories.

These aspectual verb distinctions, and particularly the semantics of the progressive tenses, lead Dowty to espouse a theory of interval semantics, earlier proposed by Bennett and Partee (1972), wherein truth conditions are given relative to an interval, rather than to a

moment, of time. Unlike other proposed changes to Montague's PTQ analysis of English, this proposal causes major modifications to the most basic semantic notion of IL, and indeed of most other temporal logics that have been studied (e.g., Rescher and Urquhart 1971). We are not convinced of the necessity of taking this step (indeed, Dowty himself says that "it results in a system that is really too powerful for natural language semantics" (Dowty 1979), p. 138). Certainly from the perspective of database querying, the complications that it introduces into the logic seem unnecessary.

The existence of an actual historical database as the heart of our logical model is the major constraint within which all of our work must be undertaken. This "given," which in essence already takes a stand on the semantics of the real world, stands as the major difference between Dowty's enterprise and ours. The semantic theory that we present is a theory of the semantics of English when used as a database query language for an HRDB, and not when used in ordinary discourse, whatever that might be. If these two theories diverge, it should be neither surprising nor disturbing, and it should be of interest to compare and contrast them.

It is apparent that an HRDB is a gross abstraction of the real world: entities are represented by unique identifiers, complex relationships are reduced to simple tuples in relations, time is rather crudely represented as a set of states of gross, perhaps even somewhat amorphous, granularity. And yet in spite of these limitations these databases are found to be useful to a large and growing number of people. What kind of constraints does the abstraction of real-world semantics embodied in the HRDB impose upon our enterprise? Precisely this: the historical database embodies a semantics that is based upon the notion of truth with respect to a state. Every fact in the database is recorded with respect to a state that "time stamps" it; this is interpreted as asserting that the fact is true *at that state*. If intervals come into play, they can be modeled as sets of time points, since time in our database view is discrete.

The differences that Dowty examines certainly exist; they do not, however, seem relevant in most database applications. Unlike Dowty, our analysis treats the progressive tenses synonymously with their simple counterparts:

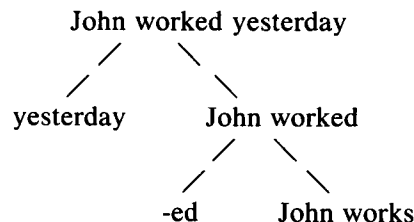
Did John earn 30K last week? *and* Was John earning 30K last week?

Does Peter work for the toy department? *and* Is Peter working for the toy department?

Perhaps it is the case that database applications do not lend themselves to handling achievements or accomplishments, but instead record stative information. In any case it is difficult to conceive of many real database examples where distinctions of the sort that Dowty's analysis is concerned with actually make sense.

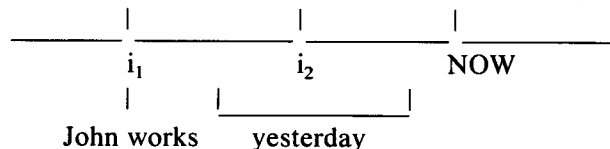
Dowty's analysis of the interaction of tenses and temporal expressions accords exactly with our own. Sentences with such interaction, such as "John worked

yesterday," cannot be analyzed as resulting from two separate temporal operators ("-ed" and "yesterday") acting on the proposition that John works, as the following example should make clear:



John works \rightarrow work'(i)(John)
 -ed \Rightarrow $\lambda p \exists i_1[[i_1 < i] \wedge p(i_1)]$
 John worked \Rightarrow $\lambda p \exists i_1[[i_1 < i] \wedge p(i_1)](\lambda i \text{ work}'(i)(\text{John}))$
 $\rightarrow \exists i_1[i_1 < i] \wedge \text{work}'(i_1)(\text{John})]$
 yesterday \Rightarrow $\lambda p \exists i_2[\text{yesterday}'(i_2) \wedge p(i_2)]$
 John worked yesterday \Rightarrow $\lambda p \exists i_2[\text{yesterday}'(i_2)$
 $\wedge p(i_2)](\lambda i \exists i_1[[i_1 < i] \wedge \text{work}'(i_1)(\text{John})])$
 $\rightarrow \exists i_2 \exists i_1[\text{yesterday}'(i_2) \wedge [i_1 < i_2] \wedge \text{work}'(i_1)(\text{John})]$

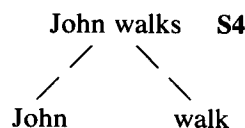
This analysis² (or the reverse, which would first apply "yesterday" and then "-ed") causes the two time operators to compete with each other, placing the event in the wrong time frame. For example, the following time line is consistent with this logical analysis, but inconsistent with the intended meaning of the English:



Instead the two temporal operators must be treated as operating in conjunction; the English -ed is, in a sense, semantically superfluous in the presence of the other time indicator. Thus the fragment has rules for applying tense operators, and separate rules for applying tense operators in conjunction with other temporal adverbials. These rules differ slightly from Dowty's in that we treat all temporal operators as operating on entire clauses, rather than simply on verb phrases. The next section will explore some of the reasons for this decision.

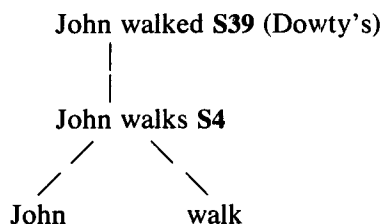
3.3.2 SENTENTIAL VS. VERB-PHRASAL TEMPORAL OPERATORS

Our analysis of tense differs from the PTQ analysis and the one in Dowty in the manner in which tense is incorporated into an English sentence. In PTQ, the Rule S4 combines a term with an IV to form a present-tensed sentence:

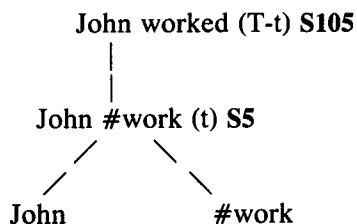


The past and future tenses are accommodated in Rule S17, which similarly combines the subject and predicate to form a sentence in either of these tenses.

Dowty's analysis is somewhat different. In his fragment a sentence is always formed first by using S4; if the tense is other than present, he introduces this with an additional rule that takes the present-tensed sentence as input and forms its past-tense counterpart, as in the following example:



Extensions to the PTQ fragment have had to deal with this issue of tense and how it interacts with the other components of a sentence. We agree with Dowty's basic premise that tense is really a property of the sentence (actually, clause) as a whole. This is particularly important when, as in our fragment, there are different kinds of sentences: declarative, wh-questions, yes-no-questions, and when-questions. For under a straightforward extension of the PTQ treatment the number of rules would proliferate, since separate rules would be needed for each kind and tense of the sentence formed by conjoining a term and a VP. However, under Dowty's treatment, the tense rules applied after S4 in most cases must undo the syntactic work that it has done, viz. the inflection of the verb as third-person singular present tense. (Semantically, the treatment is the same, i.e., the untensed version denotes exactly what the present-tensed version does.) This syntactic undoing is both inelegant and computationally unattractive. For this reason, we have incorporated into QE-III the additional categories of tensed sentences of each variety, and have modified S4 so that it creates an untensed sentence from a term and an IV. The strings of ultimate interest in the fragment, then, are the tensed sentences (categories T-t, T-WHQ, T-YNQ, and WHENQ). The following example from QE-III illustrates this for a simple declarative sentence.



In Section 4, when we discuss further examples of tensed sentences, particularly tensed questions and when-questions, we will discuss this issue further.

3.4 QUESTIONS

3.4.1 INTRODUCTION

Despite their obvious importance as a tool for gaining knowledge of the world, both linguists and philosophers

have historically considered interrogative sentences the poor relation of the declaratives, to which they have paid the bulk of their attention. Among linguists there is no generally accepted theory about the syntactic generation of English questions (Kuno and Robinson 1972, Pope 1976), and philosophers and logicians have until recently given little attention to the question of questions. More recently Engdahl (1986) explored the issue of constituent questions in Swedish, and proposed a semantic theory of questions similar to those of Hamblin (1973) and Karttunen (1977), which we shall discuss in Section 3.6. Groenendijk and Stokhof (1983) address the issue of the *appropriateness* of an answer in different situations, an issue outside the scope of the present work. Formal logic from its inception directed its attention to languages based upon the notion of formulas, abstractions of declarative sentences in natural languages. Only recently have logicians begun to investigate the semantics of questions in any depth and to develop formal languages powerful enough to express questions in order to carry out these investigations. Hintikka (1974) discusses a number of interesting linguistic and philosophical attempts to provide an analysis of questions.

Although Montague, too, focused his attention on a formal treatment of the syntax and semantics of declarative sentences in natural language, the framework of using a lambda calculus and the model theory of intensional logic, developed in PTQ, is rich enough to incorporate a view of natural language questions as well. In what seems to be his only published remark on the issue of questions, he says: "In connection with imperatives and interrogatives truth and entailment conditions are of course inappropriate, and would be replaced by fulfillment conditions and a characterization of the semantic content of a correct answer" (Montague 1973).

Perhaps inspired by this comment, a number of researchers have been investigating ways to incorporate a formal account of the syntax and semantics of questions within the framework of Montague Semantics. Hamblin (1973), Karttunen (1977), Bennett (1977, 1979), Hausser and Zaefferer (1978), and Belnap (1982) are perhaps the most important of these investigations, and we will discuss their work in relation to ours in the following section. Many of the aspects of our proposal have been adapted from or influenced by the work of these researchers.

Others not working within the MS framework have also made important contributions to our understanding of the issues involved. Approaching this issue from an entirely different perspective, researchers in artificial intelligence (AI) have over the years developed and implemented automatic question-answering theories and systems to varying degrees of success. These have ranged from some early experimental programs (Green et al. 1963) to database querying programs bound to a particular database domain (Woods et al. 1972 and

Waltz 1978) to some rather sophisticated DBQ systems today that are designed to be general and easily portable (Harris 1973, Hendrix et al. 1978). The research behind these systems seems to share a goal common to much of the work in AI (as distinct from cognitive science), i.e., an interest more in getting a system to "work" than in developing a formal theory that explains its behavior.

3.4.2 DATABASE QUESTIONS

As guidelines to help us judge any proposed theory of questions we have adopted a number of self-imposed criteria that any solution acceptable to us should meet.

1. It must fall within the general confines of Montague's framework: syntax and semantics defined in parallel, with the semantics of a phrase defined compositionally in terms of the semantics of its components.
2. The interpretation of questions should be closely analogous to the interpretation of queries in the relational database model. This means that their interpretation should be objects in the logical model which have direct analogs in the HRDM model described in Section 2.

In the relational database model "a query is a computation upon relations to yield other relations" (Maier 1983). This is an operational view of a database query; a denotational semantics view would hold that a query denotes a relation that is its answer, and would define just how, in fact, the query so denoted. In order to provide for the closest possible parallel between the interpretation of questions in our theory and the query semantics of HRDM, we hoped to define the semantics of our English query language in just such a way, viz., such that each query would denote the relation that is its answer with respect to the database. In other words, if a query in the relational database context denoted an n -ary relation over entities (i.e., a set of n -tuples), we felt that its expression as a question in our fragment should be defined to denote a function of type $\langle e^n, t \rangle$. As we shall see we were able to accomplish this easily and naturally not in the semantics, but by extending the framework of Montague Semantics to include a pragmatic component.

3. The theory should be computationally tractable. Because we are interested in developing a theory for natural language query systems that are ultimately implementable, this criterion leads us to direct our attention to solutions that fall within the general PTQ framework. This is because there have been successful results (Warren 1979 and Landsbergen 1981) implementing parsers and semantic interpretation routines for fragments defined within this framework, and we wanted to build upon this work as much as possible. While this work does not discuss a computer implementation of its results, an extension of Warren's PTQ parser (Warren 1979) to the QE-III

fragment has been implemented by Hasbrouck (1982).

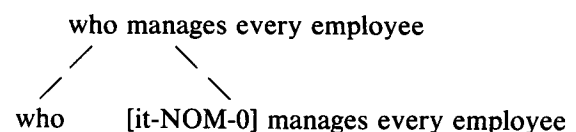
4. Proper treatment must be given to the interaction of questions and quantifiers. The PTQ treatment successfully accounts for multiple readings of sentences with interacting quantifiers ("A woman loves every man.") Our solution should likewise allow for all of the readings of questions involving quantified terms ("Who manages every employee?").
5. Y-N questions, wh-questions involving "who" and "what," and temporal questions ("when") should be provided for. This means that we do not treat indirect questions ("Tell me whether . . ."), since these do not generally arise within the database framework and could nevertheless easily be paraphrased as direct YNQs.
6. The theory should account for multiple wh-questions (e.g., "Who sells what to whom?") as these seem indispensable in a database context.

The problem of providing a correct analysis of questions that involve quantified terms is illustrated by a query like "Who manages every employee?" An analysis should only be considered adequate if it is able to find such a query ambiguous between an interpretation of "every" as "all" and also as "each." In PTQ Montague provided a solution to the familiar problem of the multiple readings of such sentences as "A woman loves every man." Under one reading there is a single woman who (magnanimously) loves each and every man, while under the other reading there is, for each man, some woman or other who loves him. A similar problem arises with respect to the interaction between ordinary and question terms, as in "Who manages every employee?"

Under one reading the questioner wishes to know what individual(s) manage all of the employees, whereas under the other reading what is wanted is really a set of ordered pairs, viz., for each employee, the set of individuals who manage him/her. Our interpretation of English questions must permit both readings, since either one is possible; the problem of disambiguating between the two is best left, as in PTQ, to a later stage that has access to domain-dependent meaning postulates.

In order to get these readings, we propose making a change in the standard interpretation of the English word "every". It is well known that this word is ambiguous—in some cases it means "all" and in others "for each". This is precisely the ambiguity in this case, and we must provide for both readings.

The first reading, where "who" has wider scope than "every," presents no problems.



[it-NOM-0] manages every employee \implies
 $\forall x[\text{EMP}'(i)(x) \rightarrow \text{MGR}'(i)(x_0) \wedge \text{manage}'(i)(x_0, x)]$
 who \implies $\lambda P \exists y[y(i) = u \wedge P(i)(y)]$
 who manages every employee $\rightarrow \exists y[y(i) = u$
 $\wedge \forall x[\text{EMP}'(i)(x) \rightarrow \text{MGR}'(i)(y) \wedge \text{manage}'(i)(y, x)]]$

The other reading requires the opposite scoping.

Who manages every employee

/ \

every employee who manages [it-ACC-O]

as well as a different meaning for "every employee." The desired reading is accomplished by allowing "every" to be ambiguous between its standard meaning "for all" and its interrogative meaning of "for each", in which it is essentially synonymous with "which".

With this treatment of "every" we obtain the desired second (and more likely) reading.

who manages [it-ACC-O] $\rightarrow \exists y[y(i) = u \wedge \text{MGR}'(i)(y)$
 $\wedge \text{EMP}'(i)(x_0) \wedge \text{manage}'(i)(y, x_0)]$
 every employee $\rightarrow \lambda P \exists z[z(i) = v \wedge \text{EMP}'(i)(z) \wedge P(i)(z)]$
 who manages every employee?
 $\rightarrow \lambda P \exists z[z(i) = v \wedge \text{EMP}'(i)(z) \wedge P(i)(z)]$
 $(\lambda i \lambda x_0 \exists y[y(i) = u \wedge \text{MGR}'(i)(y)$
 $\wedge \text{EMP}'(i)(x_0) \wedge \text{manage}'(i)(y, x_0)])$
 $\rightarrow \exists z \exists y[z(i) = v \wedge \text{EMP}'(i)(z)$
 $\wedge y(i) = u \wedge \text{MGR}'(i)(y) \wedge \text{manage}'(i)(y, z)]$

which denotes a set of ordered pairs $\langle u, v \rangle$ such that u manages v .

The problem of multiple wh-questions has a rather simple solution if one is willing to restrict one's attention to questions that involve only one wh-word; it is well known, however, that multiple wh-words require a considerably more complex treatment if the semantics is to be defined compositionally as in a Montague framework it must (Kuno and Robinson 1972). Furthermore, within the database context a restriction to single wh-questions would be too severe a constraint—it would limit the language to queries that return relations over only a single attribute.

We will discuss a number of different possible solutions to this issue of multiple wh-questions and ultimately adopt one as our solution. We will see, however, in the course of this presentation, that there are considerable technical difficulties in defining the semantics in such a way as to get it all to come out right for both single and multiple wh-questions. The solution that we adopt, involving the addition of a formally specified pragmatics for the fragment, does have this property in addition to meeting our other criteria; moreover, the simplicity of our solution, as contrasted with the considerable complexity in other proposals for a question semantics, e.g., Bennett (1977, 1979) and Hausser and Zaefferer (1978), makes it especially attractive. However, it is clear that many researchers have found the same kinds of difficulties in extending Montague's work in the direction of interpreting questions, and that

further work in this area is needed. We hope that our proposal to treat the answering of a question as a component of a formally specified pragmatics of the language, apart from its semantics, is a step in the proper direction.

3.5 THE QE-III THEORY OF QUESTIONS

3.5.1 INTRODUCTION

We first present a general view of the substance of our theory of the interpretation of questions and then discuss how this theory is carried out technically for the various types of questions that we consider. Our goal is a formal interpretation of questions as the set of their correct answers with respect to an index and a model (state and database.) This viewpoint is inspired by the relational database querying paradigm, wherein a query denotes the relation that is its answer in the current state of the database. It will be important to keep in mind the distinction between objects in a model for IL_s and objects in the relational database model. In the relational model, particularly when dealing with the relational algebra, one tends to think of all relations as being the same kind of object. One projects and joins relations at will, since these relational operators are defined generically. However, models for IL_s are strongly typed: considerations of the domains and ranges of functions are of critical importance. Within IL_s , e.g., a one-place relation of individuals is a function from D_e to D_t (denoted by expression of type $\langle e, t \rangle$), a two-place relation of individuals a function from D_e to functions from D_e to D_t (denoted by expressions of type $\langle e, \langle e, t \rangle \rangle$), etc. Thus, under our theory, a question such as "Who manages Peter?" is pragmatically interpreted (in a sense to be made clear below) as an object of a completely different type from the interpretation of a question such as "Who manages whom?" Later on we will see that this theory does not fall within the mainstream of the logical theories for question semantics that have been proposed.

3.5.2. YES-NO QUESTIONS

A semantic analysis of yes-no questions (YNQs) that meets the criteria set forth in the introduction to this section is not difficult to obtain. Since we want to interpret YNQs as either "yes" or "no" (or equivalently T or F, or 1 or 0), they can be defined to denote objects in $\{0, 1\}$. But this is just the denotation set of the corresponding declarative sentence that expresses the proposition that the YNQ asks. Thus we easily meet our criteria by providing that a YNQ denote the same proposition as that denoted by the declarative sentence from which it was derived. For example,

3-1. John manages the shoe department.
 would roughly be translated as: $\text{manage}'(i)(\text{John}, \text{shoe dept.})$

This formula is true with respect to a state s just in case John manages the shoe department in that state. Our analysis of the corresponding question "Does John

manage the shoe department?" provides that it is derived syntactically from "John manages the shoe department" and that semantically it denotes the same object in the model. The pragmatic interpretation of this question is represented by the formula: $\text{manage}'(\text{now})(\text{John}, \text{shoe dept.})$, which in effect "questions" the model as to its truth or falsity in the same way that a YNQ questions the database for the response "yes" or "no." This analysis is provided by the following pair of syntactic and semantic rules for our fragment, and by the pragmatic rules to be introduced in Section 3.5.6:

S101. (YNQ Formation)

$\langle F_{101a}, \langle t \rangle, \text{YNQ} \rangle$ and $\langle F_{101b}, \langle t \rangle, \text{YNQ} \rangle$

$F_{101a}(\theta) = \#AUX\theta^*$ where θ^* is θ with the "first verbs" unmarked.

$F_{101b}(\theta) = \text{"Is it the case that" } \theta$

T101. $F_{101a}(\theta)$ and $F_{101b}(\theta) \implies \theta'$

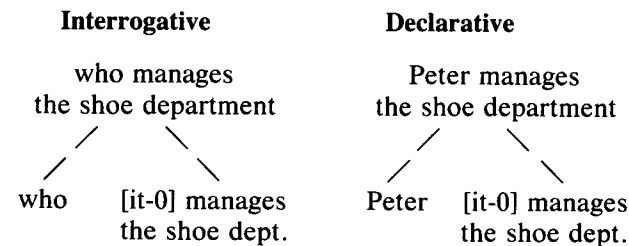
This \implies notation is used in each translation rule that is not an instance of the general rule of function application. In this case it indicates that the translation of the expression formed by performing the operation F_{101a} (or F_{101b}) on the input string ϕ is exactly the same as the translation that has already been assigned to ϕ , which we denote with the notation ϕ' . This semantic account works, since we want the interpretation of the yes-no question to be the same as the interpretation of the declarative sentence from which it is derived.

In what follows we examine the more difficult problem of defining compositionally a model-theoretic semantics for general wh-questions.

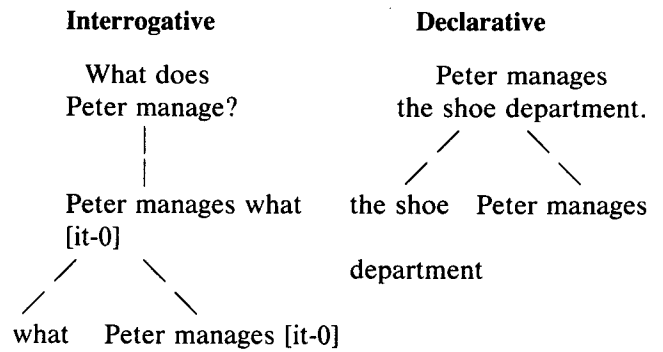
3.5.3 WH-QUESTIONS

We first present a semantic solution that does provide for a successful interpretation for questions involving only one wh-word, e.g., "Who manages Peter?" This solution has its simplicity to recommend it, but is unfortunately unable to accommodate multiple wh-questions. We then examine a number of alternative solutions to illustrate some of the many problems involved in attempting to accommodate these multiple questions.

It is obvious linguistic fact that question words like "who," "what," "whom," etc. behave syntactically in much the same way as terms like "Peter" or "an employee" (e.g., Hamblin 1973). In subject position there is virtually no difference.



while in object position there is so-called wh-Q-movement:



Nonetheless in both cases the question word takes the role of a term syntactically. Semantically, however, there is a difference. Whereas ordinary terms refer to an individual or a set of individuals in the model (actually in PTQ to the set of properties of these individuals, but this point need not concern us here), question terms seem in some way to refer to the set of all possible individuals that, when substituted for the individual variable in a matrix, makes the resulting formula true. This observation leads to the consideration that question words should denote, not sets of properties of sets of individuals, but rather functions from sets of properties to sets of individuals that have those properties. The following table makes this analogy clearer.

Term	Logical Translation	Logical Type
Peter	$\lambda P \exists x [x(i) = \text{Peter} \wedge P(i)(x)]$	$\langle \langle s, \langle \langle s, e \rangle, t \rangle \rangle, t \rangle$
an employee	$\lambda P \exists x [\text{EMP}(i)(x) \wedge P(i)(x)]$	"
who	$\lambda P \lambda u \exists x [x(i) = u \wedge P(i)(x)]$	$\langle \langle s, \langle \langle s, e \rangle, t \rangle \rangle, \langle e, t \rangle \rangle$

Because of the similarity of these wh-words to ordinary terms, both syntactically and semantically, we shall refer to them as wh-terms. The schematic essentials of the translations of two of the above examples will show how this analysis of the semantics of wh-terms provides the desired analysis of the wh-question.

Peter $\implies \lambda P \exists x [x(i) = \text{Peter} \wedge P(i)(x)]$
[it-NOM-0] manages the shoe department \implies manage'(i)(x₀, shoe-dept)
Peter manages the shoe department \implies
 $\lambda P \exists x [x(i) = \text{Peter} \wedge P(i)(x)] (\lambda i \lambda x_0 \text{manage}'(i)(x_0, \text{shoe-dept}))$
 $\rightarrow \exists x [x(i) = \text{Peter} \wedge \text{manage}'(i)(x, \text{shoe-dept})]$
who $\implies \lambda P \lambda u \exists x [x(i) = u \wedge P(i)(x)]$
who manages the shoe department \implies
 $\lambda P \lambda u \exists x [x(i) = u \wedge P(i)(x)] (\lambda i \lambda x_0 \text{manage}'(i)(x_0, \text{shoe-dept}))$
 $\rightarrow \lambda u \exists x [x(i) = u \wedge \text{manage}'(i)(x, \text{shoe-dept})]$

The first example demonstrates the PTQ-like analysis of a declarative sentence translating into a formula whose interpretation in the model with respect to a given state is a truth value. The second example provides an analysis of an interrogative sentence containing a single wh-term, using an analogous substitution rule. We obtain an expression of type $\langle e, t \rangle$ whose denotation with respect to an index is a set of entities, viz., the set

of entities who manage the shoe department in that state.

This analysis, unfortunately, cannot be generalized. Although it can also be made to provide the desired analysis for single wh-terms in direct or indirect object position, it will not allow for multiple wh-questions. To see why this is the case, consider what the S and T rules for the above analysis might look like.

S_{WH}.

If α is a WH-Term and ϕ is a formula, then $F_{WH-n}(\alpha, \phi)$ is a WH1-?, where $F_{WH-n}(\alpha, \phi)$ would be defined as some sort of substitution of α for the first occurrence of x_n , and the appropriate pronoun for each subsequent occurrence, as in the PTQ substitution rules.

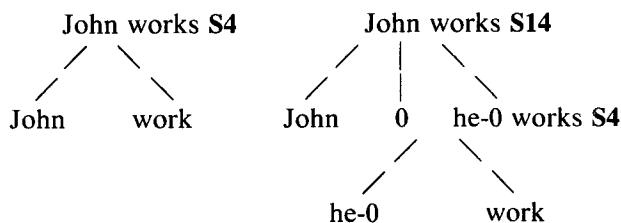
T_{WH}.

$F_{S-WH-n}(\alpha, \phi) \implies \alpha'(\lambda i \lambda x_n[\phi'])$.

Notice that this rule, unlike the analogous substitution rules in the PTQ fragment, cannot be applied recursively to its output. This is because the PTQ rules are of the form $P_\alpha + Q_\beta \implies R_\beta$ (i.e., an expression of type α combines with an expression of type β to yield another expression of type β) whereas this rule is of the form $P_\alpha + Q_\beta \implies R_\gamma$ (i.e., the output is of a different type from either of the inputs).

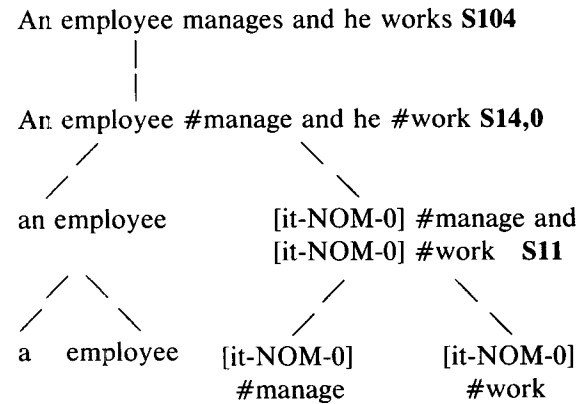
A number of alternatives present themselves at this point to allow for an analysis of multiple wh-questions within this framework. The first requires that wh-terms have different flavors ($who_0, who_1, who_2, \dots$) depending on the meaning of the expression into which they are substituted for a free variable. The second requires subcategorizing the category term, and substituting all terms in for free variables at one time. The third, and the one we have adopted, achieves the same semantic effect as the Rule T-WH, but in a two-stage process involving a separate pragmatic component. We will examine each of these ideas in turn. First, however, a word about substitution.

In the PTQ analysis, a term can become a constituent part of a sentence either by directly combining with some other constituent, or indirectly by means of substitution for a free variable that has been so directly combined. For example, consider the following two PTQ-like derivations of the sentence "John works".



Under the semantic analysis of PTQ it turns out that these two derivations receive the same translation, and hence the same "meaning." But the substitution rules are not gratuitous. They are introduced as a theory to account for pronomial co-reference and quantifier scoping. The following example illustrates how pronomial

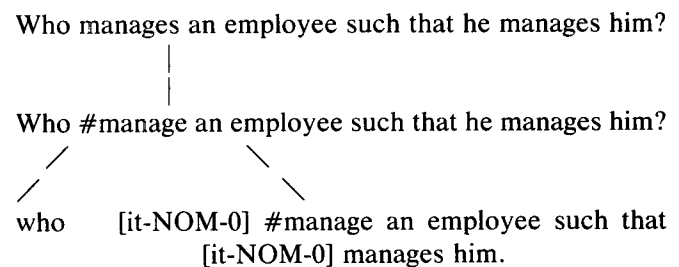
co-reference is handled by means of one of the substitution rules.



In this derivation, the Substitution Rule S14,0 provides for the reading in which the same individual is the referent of the terms "an employee" and "he". The same problem of accounting for co-reference occurs in the consideration of the semantics of questions, as the following example illustrates³.

3-2. Who manages an employee such that he manages him?

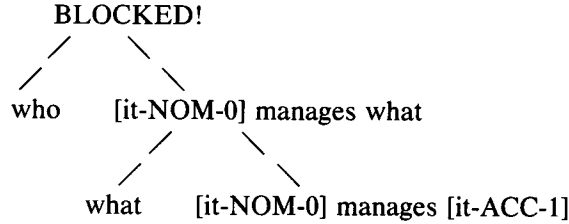
The PTQ theory of co-reference, extended to allow substitution of wh-terms, is equally able to capture the fact that "who" and "him" are co-referent, as are "an employee" and "he". Under our analysis, this sentence would be derived as follows:



This idea of extending the PTQ theory of co-reference to the case of interrogatives is not ours. It is used in most of the work on question semantics in the Montague Grammar tradition (including Karttunen 1977, Bennett 1977, 1979, and Belnap 1982). It is that theory that we have incorporated into our fragment. Because question words in our fragment are always assumed to have the entire sentence as their scope (i.e., there are no embedded question clauses), and because of the extensional nature of our theory as dictated by the database, question words can always be brought in indirectly by means of substitution rules. The difference in our respective treatments lies in our attempts to formalize the meaning given to questions.

Let us take a look now at why the analysis we have presented so far cannot be extended to multiple wh-questions. According to that analysis, the derivation of

a question like “Who manages what?” is blocked after the first wh-term is brought in.



[it-NOM-0] manages [it-ACC-1] ==>
 $\text{manage}'(i)(x_0, x_1)$ (PTQ-rules)
 what ==> $\lambda P \lambda u \exists x [x(i) = u \wedge P(i)(x)]$
 [it-NOM-0] manages what ==> $\lambda p \lambda u \exists x [x(i) = u \wedge p(i)(x)]$
 $\rightarrow \lambda u \exists x [x(i) = u \wedge \text{manage}'(i)(x_0, x)]$

Syntactically the derivation is blocked because the proposed Rule S-WH only allows a wh-term to combine with a string in the category sentence, and under the analysis “[it-NOM-0] manages what” is not a sentence. More to the point is the semantics. “Who” denotes a function from sets of properties to sets of individuals (having those properties), and the meaning of “[it-NOM-0] manages what” is not an appropriate argument for such a function.

But suppose that the “who” that combined with formulas to form expressions denoting sets of individuals were a different function from the “who” that combined with expressions denoting sets of individuals to form expressions denoting sets of ordered pairs of individuals, etc.? Suppose, that is, that the English “who” were really a syntactic realization of a number of different meanings, who_0 , who_1 , etc., as follows:

who_0 combines with propositions to form a set of individuals,

who_1 combines with sets of individuals to form a set of ordered pairs, and in general,

who_i combines with sets of ordered i -tuples to form sets of ordered $i+1$ -tuples

These different functions of the English “who” would be captured by their different translations into the logic (reflecting their interpretation as different semantic functions).

who-word	Translation	Type
who_0	$\lambda P \lambda u \exists x [x(i) = u \wedge P(i)(x)]$	$\langle\langle s, \langle s, e \rangle, t \rangle, \langle e, t \rangle \rangle$
who_1	$\lambda R \lambda v \lambda w \exists z [z(i) = v \wedge R(i)(z, w)]$	$\langle\langle s, \langle\langle s, e \rangle, \langle e, \rangle \rangle \rangle, \langle e, \langle e, t \rangle \rangle \rangle$

With this analysis we could complete the above derivation, previously blocked, as follows:

[it-NOM-0] manages what ==>
 $\lambda u \exists x [x(i) = u \wedge \text{manage}'(i)(x_0, x)]$
 who (as who_1) ==> $\lambda R \lambda v \lambda w \exists z [z(i) = v \wedge R(i)(z, w)]$
 who manages what ==> $\lambda R \lambda v \lambda w \exists z [z(i) = v \wedge R(i)(z, w)] (\lambda i \lambda x_0 \lambda u \exists x [x(i) = u \wedge \text{manage}'(i)(x_0, x)])$

$\rightarrow \lambda v \lambda w \exists z \exists x [z(i) = v \wedge x(i) = w \wedge \text{manage}'(i)(z, x)]$

In theory there would be an infinite number of such (related) meanings to the word “who,” one for each natural number n , and we could even give a rule for generating these meanings inductively from the single meaning of who_0 . In practice (and computationally), since ordinary English (and even “database-ese”) allows for only a small number of terms in only a small number of places (subject, direct and indirect objects, object of preposition, “list. . .” requests, etc.) only a small number would actually ever be used in any normal English question. The S and T rules for this analysis would be something like the following:

S_{WH-n}
 If $\alpha \in P_{WH-Term-i}$ and $\beta \in P_{?i?}$ (i.e. β denotes a set of i_1 -tuples), then $F_{WH,n}(\alpha, \beta) \in P_{?i?}$, where $F_{WH,n}(\alpha, \beta)$ is the result of replacing the first occurrence of [it-CASE- n] in β with α , and replacing all subsequent occurrences of [it-CASE- n] in β with he/she/it or him/her/it, respectively, according to the gender of α and the CASE of [it-CASE- n].

T_{WH-n}
 $F_{WH,n}(\alpha, \beta) ==> \alpha'(\lambda i \lambda x_n \beta')$
 Moreover, to account for derived wh-terms like “which employee” in “which employee sells shoes?”, we could extend this analysis to the interrogative determiners “which” and “what”. This would dictate that $which_0$ combined with employee to form $[which\ employee]_0$, $which_1$ with employee to form $[which\ employee]_1$, etc., of the appropriate types.

This analysis, while inelegant, is not really so far-fetched. After all, in asking, “Who manages John?” “who” is in some way asking for a set of individuals, viz., those that manage John. In asking, “Who manages whom?” however, rather than asking for a set of individuals, “who” is asking in conjunction with “whom” for a set of ordered pairs such that the first component manages the second component. A theory such as the above sketch would claim that English allows for these many semantic functions of interrogative terms to be performed by the same surface words like “who”.

We might also point out here a closely related alternative to this approach. Instead of having an infinite number of meanings for each wh-term, we could suffice with one and allow an infinite number of syntactic and semantic rules for performing the substitutions. These rules would perform the necessary conversions of the meanings, not of the wh-term, but of the sentential form into which it is being substituted. Thus, e.g., the T-WH-1 rule for combining “who” with “[it-NOM-0] manages whom” would form the following expression (where WHO* stands for the translation of “who”):

$\lambda w [WHO*[\lambda i \lambda x_0 \beta^*(w)]]$

For example, combining “who” with “[it-NOM-0] manages what”:

$$\begin{aligned} \text{who} & \implies \lambda P \lambda v \exists y [y(i) = v \wedge P(i)(y)] \\ \text{[it-NOM-0] manages what} & \implies \lambda u \exists x [x(i) = u \wedge \text{manage}'(i)(x_0, x)] \\ \text{who manages what} & \implies \lambda w [\lambda P \lambda v \exists y [y(i) = v \wedge P(i)(y)]][[\lambda i \lambda x_0 [\lambda u \exists x [x(i) = u \wedge \text{manage}'(i)(x_0, x)]](w)]] \\ & \rightarrow \lambda w [\lambda P \lambda v \exists y [y(i) = v \wedge P(i)(y)]][[\lambda i \lambda x_0 [\exists x [x(i) = w \wedge \text{manage}'(i)(x_0, x)]]]] \\ & \rightarrow \lambda w \lambda v \exists y \exists x [y(i) = v \wedge x(i) = w \wedge \text{manage}'(i)(y, x)] \end{aligned}$$

Notice that this rule schema essentially converts the one-place relation denoted by one of its arguments (“[it-NOM-0] sells what” in the example) into a formula (by function application to the new individual variable w) in order to allow the single meaning of “who” to apply. Lastly, it λ -abstracts this variable w over the result in order to obtain a two-place relation. A slightly unfortunate result of this rule is that the order of the individuals in the relation is exactly opposite from the order in which the wh-terms were quantified.

A second possible approach that would handle multiple wh-questions would dispense with this essentially inductive treatment of i-place questions and attack the problem all at once. Such a theory would derive all questions in the same manner, by simultaneously substituting all wh-terms into the matrix sentence, keeping track of which terms were substituted for which variables. For example, the question “Who supplies what to which departments such that they sell shoes?” would be analyzed as follows:

“Who supplies what to which departments such that they sell shoes?”

who:0	[it-NOM-0] supplies [it-ACC-1] to [it-DAT-2]
what:1	such that [it-NOM-2] sells shoes
which department:2	

Either of these two basic theories is possible; we have rejected them both for a number of reasons. First, the use of an infinite number of meanings for wh-terms and wh-determiners (or an infinite number of rules schemas for their substitution) requires the same technique for each of the tense rules, and for each of the tense rules with time adverbials, and for each of the when-question rules, the rules for when-questions with tenses, for when-questions with tenses and time adverbials, etc. In other words, accepting a solution that types all questions differently depending upon what they ask for forces the inclusion of rule schemas for all of the other semantic functions that in a simple theory would operate only on one type, the type given to sentences. (Later we will discuss how the solutions of Bennett (1977, 1979), Belnap (1982), and of Hausser and Zaefferer (1978) entail a similar rippling effect of complexity

throughout the rest of the semantic theory already developed for declarative sentences.)

For an example of this effect in the theory under consideration, consider what the rule for adding past tense to a sentence would look like. (Recall our arguments for the necessity of treating tense as a property of the entire sentence.) Such a rule would have to be of the form $P_\alpha + \text{-ED} \implies Q_\alpha$, where α could be the category declarative sentence (type t), 1-term question (type $\langle e, t \rangle$), 2-term question (type $\langle e, \langle e, t \rangle \rangle$), etc. Because of the strict typing system of IL_s (and of the categorial grammar of the PTQ theory of English syntax), this would require an infinite number of such rules, one for each of the possible input categories. While such a scheme is possible, it seems to violate a concern for simplicity and elegance.

An additional problem with a theory dependent upon simultaneous substitution is a difficulty of conceiving of it in semantic terms. While the translation rules for such a theory can probably be described (they would be somewhat complicated), they strongly suggest the view that the translation rules themselves are the semantics, when in fact they are nothing more than syntactic operations on strings of logical symbols. (This is a common problem for people working with Montague Grammars, occasioned by the indirect way that the semantics for English is specified. Dowty (1978) refers to this problem when he reminds us that “the translation is a completely dispensable part of the [PTQ] theory. The ‘real’ semantic interpretation of an English sentence is the model-theoretic interpretation of its translation and nothing but the model-theoretic interpretation of that translation.”) When examined in terms of the semantic space of functions in the model, it is not clear what simultaneous substitution in the syntax denotes model theoretically.

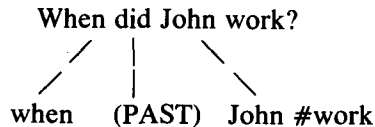
3.5.4 TEMPORAL QUESTIONS

When-questions are different from any of the questions we have considered so far for three reasons. First, they ask about an object of a different logical type: all of the questions we have considered have been treated as in some way referring to sets of n-tuples of individuals (of type e); when-questions, on the other hand, refer to states (of type s).

Second, although sentences can and do make reference to more than one time (“I know that John was here”), multiple when-questions are very infrequent. In most situations “when” in English can generally only be asked once in a given sentence. “When and when . . . ?” does not make sense, and questions like “When did John come and when did he leave?” are really two conjoined questions.⁴ We have not accounted for multiple when-questions of these sorts in our theory. An interpretation for them could be formulated in a manner analogously to our treatment of multiple wh-questions, but this would require modifying the treatment of time (modeled after Montague’s) in the semantics. We will

have more to say about this in our discussion of pragmatics.

Finally, when combining with *wh*-terms, “when” must be brought in last to have the widest scope, for essentially the same reasons that led to the recognition that tense had to have widest scope. Moreover, some account must be given of how “when” interacts with our treatment of tenses, the other major temporal indicator in the surface structure of English. The following example indicates how “when” is introduced into a sentence and captures the variable *i* in all of its free occurrences.



John #work \rightarrow EMP_{*}'(i)(John)

When did John work? \rightarrow $\lambda p \lambda i_1 [[i_1 < i]$
 $\quad \quad \quad \wedge p(i_1)] (\lambda i \text{EMP}_*'(i)(\text{John}))$
 $\quad \quad \quad \rightarrow \lambda i_1 [[i_1 < i] \wedge (\lambda i \text{EMP}_*'(i)(\text{John})(i_1)]$
 $\quad \quad \quad \rightarrow \lambda i_1 [[i_1 < i] \wedge \text{EMP}_*'(i_1)(\text{John})]$

Thus the question is interpreted as asking for the set of times in the past at which John was an employee.

3.5.5 PRAGMATICS OR SEMANTICS?

Most theories of question semantics, including those sketched above, and those of a number of other researchers in the Montague framework to be discussed shortly, make significant complications to the semantics of other parts of the PTQ analysis in order to incorporate these new sentences. Perhaps we are overtaxing the semantic component of our language theories, asking it to do for us more than it was intended to do. For example, most theories of question semantics have attempted to include some representation of the answer of the question as part of its denotation. Is there not something odd in a theory that holds that a question denotes its answer, especially if one has tried (or tried not) to think of denoting as a formal counterpart to the intuitive notion of meaning? Yet in one form or another (denoting the set of possible answers (Hamblin 1973), the set of correct answers (Karttunen 1977), functions from sequences of individuals to propositions (Bennett 1977, 1979 and Belnap 1982), and our sets of *n*-tuples of individuals) many researchers have been investigating ways to accomplish this in a formal semantics. The similarity between *wh*-terms and unbound pronouns (“Who loves whom?” versus “He loves him”) suggest another approach, viz., one in which

1. the **semantic component** provides that questions *denote* as declarative sentences (with unbound pronouns) do, and
2. the **pragmatic component** provides that questions are *interpreted* as requests for their answers.

Pragmatics is the least understood branch of the tripartite division of the study of language that Morris (1938)

proposed in his theory of semiotics. This century has seen tremendous successes in the development of formal logical syntax and model-theoretic semantics, but very little in the way of formal pragmatics. (Martin (1959) is an early attempt in this direction.) Marciszewski (1971) and more recently Levinson (1983) together contain a thorough analysis of the various approaches that have been taken to define the scope of linguistic pragmatics, from its origin in Morris’s definition of semiotics (Morris 1938) to the present day. It is obvious from these accounts that there has been less agreement as to the scope of this branch of the field. Morris (1938) originally defined pragmatics as the study of “the relation of signs to interpreters.” Later, at least partly in response to Carnap’s proposal that “if in an investigation explicit reference is made to the . . . user of a language, then we assign it to the field of pragmatics” (Carnap 1947), Morris felt the need to expand upon this definition. In Morris (1946), he says that pragmatics is “that portion of semiotic which deals with the origins, uses, and effects of signs within the behavior in which they occur.”

Montague’s conception of pragmatics (Montague 1968, 1970a, and 1973), based upon Bar-Hillel’s (Bar-Hillel 1954) discussion of indexical expressions, represents a departure from the traditional view. Hamblin (1973) felt that Montague’s incorporation of a pragmatic component directly in the syntax and semantics was unconventional, and felt the need “to defend pragmatics from this weakened interpretation . . . Pragmatics is the study of the use (not just reference) of language of all kinds; or, if it is not, we need a new name for the study that complements syntax and semantics. Montague’s ‘pragmatics’ would be better classed as a special part of semantics.” Dowty (1978), while admitting that “the linguist’s use of the term pragmatics is far from standardized,” adopts the view that it should encompass direct and indirect speech acts, presuppositions, and implicatures, and explicitly rejects Montague’s use of the term to encompass a treatment of indexical expressions.

What we propose in our theory of questions is that the proper place for considering the answer(s) to a question is in a separate theory of pragmatics for the language. We do not propose a completely general theory of pragmatics. But we believe that incorporating into our fragment a formal pragmatic component that treats the notion of a response to a question is defensible as at least one component of a **theory of language use**. In the first place, Montague notwithstanding, it falls within the confines of pragmatics as that term is generally understood. For whether one speaks of “the use and effects” of language (Morris 1946), the “relation of signs to their interpreters” (Morris 1938), the notion of “speech acts” (Dowty 1978), or the “linguistic means for effecting literal purposes” (Kasher 1977), it is clear that the notion of responding to a question is encompassed. Our attempt to formalize a pragmatic compo-

ment to QE-III accords well with what Stalnaker (1972) sees as the goals of “a formal semiotics no less rigorous than present day logical syntax and semantics.” Those goals, he goes on to say, include an analysis of such linguistic acts as “assertions, commands, . . . , requests . . . to find necessary and sufficient conditions for the successful (or perhaps in some cases normal) completion of the act.”

A second argument in favor of this approach comes from looking at the way that linguists have described the concept of a question. Linguists have traditionally classified sentences into four distinct types: declarative, interrogative, imperative, and exclamatory. A glance through some standard text or reference books on English grammar reveals two separate approaches to the rationale behind this scheme. According to one school, as in Roberts (1954), it is based upon the “different kinds of meaning” a sentence may have. The other school, as in Quirk and Greenbaum (1974), considers that the distinction is based upon such criteria as word order in the sentence, presence or absence of a subject, the presence of an interjection, etc. Clearly the disagreement is over whether to consider this a syntactic or a semantic distinction. Perhaps in some sense it is mainly a pragmatic one, reflecting both the use and effects of the utterance.

Finally, this approach in its technical details is both simple and elegant. It removes from semantics the burden of providing an account of the response to a question, and allows it to do what semantics theories have always done best—account for reference. Then, just as the semantics of a language is based upon its syntax, the pragmatics is based upon both the syntactic and semantic analyses (or, in Hamblin’s phrase, it “complements syntax and semantics”). The simplicity with which we can state the pragmatic rules for our fragment, which take into account the notion of the answer to a question, is based upon this ability to have both the syntax and the semantics at hand upon which to build a theory of pragmatics. An example should make this clear.

In QE-III, questions denote (a semantic concept) just as declarative sentences do. Thus QE-III gives the following semantic analyses for “Who manages whom?” in the syntactic category *wh*-question, and for “He manages him” in the category declarative sentence.

who manages whom? $\rightarrow \exists x[x(i) = u_2 \wedge \text{EMP}_*'(i)(u_1) \wedge \text{AS-1}(u_1, x)]$

he manages him $\rightarrow \exists x[x(i) = u_2 \wedge \text{EMP}_*'(i)(u_1) \wedge \text{AS-1}(u_1, x)]$

Both are treated as denoting the same object with respect to an index, a variable assignment, and a model. But they are interpreted differently in the pragmatics. Pragmatics in QE-III is a function that, given a derivation for an expression of QE-III together with its

syntactic category and its (semantic) denotation, returns a (possibly) new object in the same model as its pragmatic interpretation. Thus, although we view pragmatics as a separate component of a language theory, it is closely allied to the semantics—both provide interpretations of linguistic expressions within the context of the same logical model. The formal definition of the pragmatic component of IL_s results in these two sentences, interpreted pragmatically, denoting what the following expressions of IL_s denote:

who manages whom? $\rightarrow \lambda u_2 \lambda u_1 \exists x[x(i) = u_2 \wedge \text{EMP}_*'(i)(u_1) \wedge \text{MGR}'(i)(x) \wedge \text{AS-1}(u_1, x)]$

he manages him $\rightarrow \exists x[x(i) = u_2 \wedge \text{EMP}_*'(i)(u_1) \wedge \text{MGR}'(i)(x) \wedge \text{AS-1}(u_1, x)]$

The pragmatic interpretation of the question is the set of *n*-tuples that answer it, while of the declarative sentence is the same as its denotation.

3.5.6 THE PRAGMATICS OF QE-III

The pragmatics that we give here for QE-III is a simple theory of the effects of producing an expression in that language within the assumed context of a question-answering environment. That is, we assume that a user of QE-III is using the language to produce some effect within this context, and it is this effect which we formalize as the pragmatic component of the language definition. We could, of course, have defined the pragmatics in the same manner as the semantics was defined, i.e., inductively over the syntax. However, in doing so we would have seemed to be giving some status or importance to the pragmatic interpretation of expressions in every category of QE-III. Because we had no real intuition about what the pragmatic interpretation of, say, the expression “in 1978” represented, we decided upon a different form of the definition. Accordingly, our definition provides a pragmatic interpretation for expressions in any of the several sentential categories of the language, namely T-YNQ, T-WHQ, WHENQ, and T-t. (Section 3.2 contains a discussion both of some of the issues involved in our decision to present a separate pragmatic component to the formal theory of QE-III, as well as some of the considerations for the present form of this theory.)

The following preliminary definitions are needed before stating the pragmatic rules.

1. By $/\alpha \backslash$ is meant a derivation tree for the meaningful expression α of QE-III, as informally understood from our discussion of the syntax. We further assume that nodes of derivation trees are labeled with ordered triples $\langle A, B, C \rangle$, such that *A* is the meaningful expression derived at that node, *B* is its syntactic category, and *C* is the rule of syntax applied at that step in the derivation. For simplicity, we shall refer to component *A* of the root of $/\alpha \backslash$ as α , and to the component *B* as $\text{CAT}(/ \alpha \backslash)$.

2. The translation rules guarantee that corresponding to any derivation tree $/\alpha\backslash$ for $\alpha \in \text{ME}_{\text{QE-III}}$ there is a *unique* translation into IL_s . By $T(/ \alpha \backslash)$ shall be understood this unique translation, and by $[/\alpha\backslash]_M$ the denotation of $/\alpha\backslash$ provided indirectly via $T(/ \alpha \backslash)$ with respect to the model M .
3. There are two standard ways of defining a (Tarskian) model-theoretic semantics. One is to define the notion of denotation with respect to a model M only, in which case formulas, e.g., denote the set of their satisfying variable assignments. The other, and more usual, procedure is to define the denotation with respect to a model M and a variable assignment g , in which case a formula denotes either True or False. The two notions are, for all practical purposes, equivalent. Since for the purposes of pragmatics we shall want to consider that open formulas denote the set of their satisfying variable assignments, we shall in this section refer to the notion of denotation with respect to a model M only.
4. If $[/\alpha\backslash]_M$ is a function whose domain is $\text{As}(M)$, the set of all possible variable assignments over M , and if further $V = \{v_1, \dots, v_k\}$ is a set of variables of IL_s , then by $\text{II}_V([/\alpha\backslash]_M)$ is understood the restriction of $[/\alpha\backslash]_M$ to the domain V . Note that if $V = \emptyset$, then $\text{II}_V([/\alpha\backslash]_M)$ is defined to be just $[/\alpha\backslash]_M$.
5. If f is any function with domain $\text{As}(M)$, then $\text{now}(f)$ is the restriction of f to the domain $\text{As}_{\text{now}}(M)$, where $\text{As}_{\text{now}}(M) = \{g \mid g \in \text{As}(M) \text{ and } g(i) = F(\text{now})\}$, that is, that subset of the possible variable assignments for M for which the distinguished time variable i is interpreted as denoting that state denoted by the constant **now**.
6. By $\text{FV}(/ \alpha \backslash)$ we shall understand the set $\{i_1, i_2, \dots, i_n\}$ of indices of the variables (expressions of the form $[it\text{-CASE-}i]$) occurring free in α . This notion will not be defined rigorously here, but would be defined inductively over the structure of $/\alpha\backslash$ in the usual manner, with particular attention paid to which rules bind occurrences of variables (all of the PTQ substitution rules) and which rules leave them free (e.g., the rules that introduce wh-terms.) This definition would be analogous to the definition of the set FV_e of variables of type e occurring free in a logical expression, in particular in the expression $T(/ \alpha \backslash)$. It is clear that if $\text{FV}(/ \alpha \backslash) = \{i_1, \dots, i_n\}$ then $\text{FV}(T(/ \alpha \backslash)) = \{u_{i_1}, \dots, u_{i_n}\}$. However, we emphasize that $\text{FV}(/ \alpha \backslash)$ is defined over the derivation tree of α (i.e., over the syntax of QE-III) and make no reference to the (intermediate) translation of this tree into IL_s .
7. Finally, if β is a meaningful expression of IL_s , and if the free variables of type e in β , $\text{FV}_e(\beta) = \{u_{i_1}, u_{i_2}, \dots, u_{i_n}\}$, are such that $u_{i_1}, u_{i_2}, \dots, u_{i_n}$ are in alphabetical order, then $\text{LC}_{\text{FV}_e}(\beta)$ is the unique expression: $\lambda u_{i_1} \dots \lambda u_{i_n} \beta$ formed by first prefixing

β with λu_{i_1} , then prefixing λu_{i_2} to the result, and so on.

In order to understand the form of some of the following definitions we state the following fact (the proof follows directly from the translation rules of QE-III): If β is the translation of any meaningful expression α of QE-III, then the free variables of α are all of type e , except for the possible exception of the distinguished variable i of type s .

The rules of pragmatics that we now state constitute a definition of the pragmatic function, in a manner analogous to the way in which the translation rules constitute a translation relation. In particular they constitute a definition of the function $P: P: / \text{QE-III} \backslash \rightarrow M \cup \{\text{ERROR}\}$, which assigns to any derivation tree of a meaningful expression α of QE-III, either an object in the model M or the distinguished symbol "ERROR" as its pragmatic interpretation.

P1. If $\text{CAT}(/ \alpha \backslash) \notin \{\text{WHENQ}, \text{T-WHQ}, \text{T-t}, \text{T-YNQ}\}$

then $P(/ \alpha \backslash) = \text{ERROR}$.

P2. If $\text{CAT}(/ \alpha \backslash) \in \{\text{WHENQ}, \text{T-WHQ}, \text{T-t}, \text{T-YNQ}\}$ then $P(/ \alpha \backslash) = \text{II}_{\text{FV}_e}(\text{now}([/\alpha\backslash]_M))$

Rule P1 ensures that only sentences have a pragmatic interpretation. Rule P2 ensures that all sentences are interpreted with respect to the "current" state index, and that in the case of questions, the infinite sequences of variables that the question denotes is projected down to include only the questioned variables.

It is clear that the set of sequences given by $\text{II}_{\text{FV}_e}(\text{now}([/\alpha\backslash]_M))$ is equivalently represented by the denotation of the expression $\text{LC}_{\text{FV}_e}(\lambda i T(/ \alpha \backslash)(\text{now}))$ of IL_s with respect to M and g . P2 is therefore alternatively defined as:

$$P(/ \alpha \backslash) = [\text{LC}_{\text{FV}_e}(\lambda i T(/ \alpha \backslash)(\text{now}))]_{M,g}$$

What this alternative definition allows us to do is to utilize the semantic notion of denotation to define the pragmatic interpretation of sentences in QE-III. For it allows us to take a translation $T(/ \alpha \backslash)$ of any sentence α and determine its pragmatic interpretation as the denotation of the expression $\text{LC}_{\text{FV}_e}(\lambda i T(/ \alpha \backslash)(\text{now}))$ and thus evaluate the pragmatic interpretation of α in terms of the semantics of IL_s by means of this simple syntactic transformation on $T(/ \alpha \backslash)$.

3.5.7 CONCLUSIONS

The QE-III theory defines the denotation of a question in exactly the same way as the denotation of the corresponding declarative sentence that has pronouns in place of the interrogatives, but defines its pragmatic interpretation as the set of n -tuples that answer it. We have discussed our initial attempts to accomplish this result directly, by having wh-terms denote functions from sets of properties to sets of individuals that had those properties. Technically, we discovered that to

accomplish this directly required a considerable complication of the semantics throughout the structure of our fragment. And we discovered, as we shall see, that others with similar goals had also been forced to introduce more complexity into their logical model in order to accomplish these goals in the semantic component of their theory. Finally, we realized that by eliminating as a goal of the semantics the capturing of the answer(s) of questions, we could leave the basic semantic theory of PTQ intact, and moreover, we could easily accomplish this goal in the pragmatics.

This concludes our informal discussion of the syntax, semantics, and pragmatics of QE-III. We now proceed to discuss the theory in relation to some of the other work in the field of Montague Semantics which has attempted to extend the PTQ fragment to include a theoretical account of the syntax and semantics of questions.

3.6 RELATED WORK

3.6.1 INTRODUCTION

Two common threads run through much of the recent work on formalizing a theory of questions. The first is the idea that all questions should be defined so as to denote objects of the same type. Generally, this has meant propositions or sets of propositions, but it seems that even before the choice of just what questions denote was made, this "single semantics" viewpoint had been adopted. The other, as we have already pointed out, is that some account of the answer(s) to a question should be included at least as a component of its semantics. When combined with other factors these two biases have led to somewhat different results. Thus Hamblin (1973) suggests that a question denotes the set of all "propositions that count as answers to it"; Karttunen (1977), "the set of propositions expressed by[its] true answers"; Bennett (1977, 1979) and Belnap (1982), who worked with Bennett on the theory, "sets of open propositions: functions from sequences of individuals to propositions."

3.6.2 KARTTUNEN

As we have said, Karttunen (1977) presents an analysis of the semantics of questions that falls within the single semantics tradition. (Hamblin (1973) earlier proposed a treatment similar to Karttunen's, but his theory was not worked out in as much detail.) In Karttunen's theory, for example, the question "Who manages John?" would roughly be translated as: $\lambda p \exists x [p(i) \wedge p = \lambda i [\text{manage}'(i)(x)(i), \text{John}]]$. Semantically, this expression, when interpreted with respect to a model and a state, denotes the set which contains, for each person x that manages John, the proposition that x manages John.

Such a treatment of the semantics of questions seems inappropriate to us for two related reasons. First, it seems to confuse propositions with the sentences that express them. Whatever a proposition might be in our informal use of the term, it is in the formal system

defined by IL_s a function from indices to truth values, or equivalently a set of indices. In order to see why this seems an inappropriate choice for the semantic object denoted by a question, consider a model in which the constants *manage'* and *love'* are interpreted as follows:

$$\begin{aligned}
 F(\text{manage}') &= \begin{array}{l} | 1978 \rightarrow \{ \langle \text{Mary}, \text{John} \rangle, \langle \text{Susan}, \text{John} \rangle \} | \\ | 1979 \rightarrow \{ \langle \text{Mary}, \text{John} \rangle, \langle \text{Bill}, \text{John} \rangle \} | \\ | 1980 \rightarrow \{ \langle \text{Bill}, \text{John} \rangle \} | \end{array} \\
 F(\text{love}') &= \begin{array}{l} | 1978 \rightarrow \{ \langle \text{Bill}, \text{John} \rangle, \langle \text{Susan}, \text{John} \rangle \} | \\ | 1979 \rightarrow \{ \langle \text{Bill}, \text{John} \rangle \} | \\ | 1980 \rightarrow \phi | \end{array}
 \end{aligned}$$

Now consider the two queries Q1: "Who manages John in 1978?", which translates to $\lambda p \exists x [\text{manage}'(1978)(x, \text{John}) \wedge p = \lambda i \text{manage}'(i)(x, \text{John})]$; and Q2: "Who loves John in 1978?", which translates to $\lambda p \exists x [\text{love}'(1978)(x, \text{John}) \wedge p = \lambda i \text{love}'(i)(x, \text{John})]$. Given these translations, the interpretation of these two queries in this model, [Q1] and [Q2], is:

$$\begin{aligned}
 [Q1] &= \{ \{1978, 1979\} /*Mary manages John */ \\ &\quad \{1978\} \quad } /*Susan manages John*/ \\
 [Q2] &= \{ \{1978, 1979\} /*Bill loves John */ \\ &\quad \{1978\} \quad } /*Susan loves John */
 \end{aligned}$$

Under this interpretation, both queries, which are obviously quite distinct, denote exactly the same set of propositions in the model, the set containing the proposition {1978, 1979} and the proposition {1978}. Thus under this interpretation we cannot distinguish between these two questions—they are semantically equivalent in the database under this theory.

The second and related objection is that under this interpretation all direct mention of the entities (Mary, Susan, John, . . .) involved disappears. Instead, the theory claims that the question denotes a set that contains sets of states (years). What this implies is that there is no obvious way of going backwards from these objects in the model (the sets of propositions) to some useful expression in a language (English) that names them. Since in this theory the denotation of questions loses the people involved, we have no simple way to recover their names and report them to the questioner. The theory neglects considering the use and effects of the question. Moreover, there seems to be no way even to add a pragmatic component to such a theory in order to account for a question's answer(s), for on the one hand the syntax has no mention of the names of the individuals involved, nor on the other hand does the denotation involve any individuals at all. In the pragmatics of our theory the two queries would instead be interpreted, with respect to a given database, as follows (where [Qi] now means the pragmatic interpretation of Qi): [Q1] = {Mary, Susan}, and [Q2] = {Bill, Susan}.

With these interpretations we have not lost the people involved, and there is an obvious relationship between these objects and English expressions for them

("Mary and Susan" and "Bill and Susan"), as well as the relations in the database that express the same information.⁵

Q1	Q2
Mary Susan	Bill Susan

As noted earlier, Bennett discussed the issue of the logic of questions in two separate papers, and collaborated with Belnap in the development. Their theory is presented cumulatively in Bennett (1977, 1979) and Belnap (1982). Motivated again by the goal of a single semantics, and even more strongly by a desire to account for the individuals that answer the question, Belnap and Bennett develop a theory that incorporates sequences of individuals into the model theory. Thus a question like "Whom does John love?" is treated as denoting a set of functions from sequences of individuals to propositions. Essentially all and only those sequences that close the open proposition "John loves [it-ACC-0]" and make it true are included in this denotation. What this is tantamount to is incorporating the standard (Tarskian) notion of a variable assignment into the model theory, instead of leaving it in the meta theory of the logic. For technical reasons the entire system must be altered to include these sequences, so that even sentences are no longer translated into formulas, but rather into expressions denoting sets of such sequences. This rippling effect of the complications to the semantics is extraordinarily reminiscent of the problems we had in formulating a theory with inductive wh-terms.

In order to accomplish this result, the set of types of the IL is expanded to include as a basic type n , expressions of type n denoting a natural number. Thus the natural numbers must be included as objects in the model, as well as functions constructed from them. Of particular interest in their theory are the functions from N to individuals, i.e., sequences. The ripple effect necessitates that "all expressions of English [denote] functions from sequences of individuals to their usual extensions" (Bennett 1979). Even sentences are no longer translated into formulas, but rather into expressions of type $\langle\langle n, e \rangle, t \rangle$ that denote sets of sequences. Unfortunately, the results of this complication to the logic and the English translations do not seem to justify the cost. Certainly this theory represents a step closer to the goal of capturing explicitly in the denotation the individuals that answer the question, so it is an improvement over the proposition proposal. But these individuals are hidden somewhere inside infinite sequences of individuals, with no indication of their position within those sequences.

An example should clarify this point. In order to

understand it, we provide the following table of the types of the variables used.

Variable symbols	Type of variable symbol
P	$\langle e, t \rangle$: sets of individuals
r, s	$\langle n, e \rangle$: sequences of individuals
O	$\langle \langle n, e \rangle, \langle s, t \rangle \rangle$: open propositions

In the Bennett/Belnap theory, an open sentence like "John loves him" is translated as: $[\lambda s[\text{love}'([\lambda \text{APP}(s(1))](\text{John}))]]$, which denotes (ignoring intentions) the set of sequences such that John loves the first member of each of them. The corresponding question "Whom does John love?"⁶ would be: $[\lambda O[O = [\lambda s[\text{love}'([\lambda \text{APP}(s(1))](\text{John}))]] \wedge \exists r[\text{O}(r)]]]$, which denotes a set of open propositions. But these again involve infinite sequences of individuals, and there is no indication of which projection of these sequences represents the individuals that John loves.

This problem of having the individuals that constitute the answer embedded in infinite sequences without knowing how to project them out is the same one that we have in our semantic theory. For our semantics translates questions into open formulas, which denote the set of variable assignments that satisfy the formula. Our relegating to pragmatics the task of projecting these variable assignments could also be used to solve this problem here. But if this is the case, then what is gained by paying the price for the complication to the model theory and the translation rules? This use of sequences in effect duplicates the variable assignment of their Tarskian meta-theory (albeit restricted to the domain D_e) in the object language and in the logical model with no noticeable advantages.

3.6.3 HAUSSER AND ZAEFFERER

The proposal of Hausser and Zaefferer (Hausser and Zaefferer (1978) and hereafter **H-Z**) is quite different from the other theories we have discussed, and makes a number of interesting points. The theory is motivated early in the paper by a discussion of the range of answers that are possible to any given question, and a classification of these possibilities as ranging from "minimal" to "redundant". For example, in answer to the question "Who dates Mary?" the following list of possibilities is cited:

- a. Bill.
- b. Bill does.
- c. Bill does so.
- d. Bill dates her.
- e. Bill dates Mary.

Answer (e), of course, is just what the propositional approach would say that the question denotes (assuming Mary is going steady with Bill.) H-Z goes on to say, however, that "the truth value of the answer

expression will depend on the question in the context of which it is uttered, except for [the completely redundant answer]. This shows that redundant answers are not very interesting from a semantical point of view since their semantic representation is identical to that of ordinary declarative sentences. Since both redundant and non-redundant answers are possible, and since non-redundant answers are generally much more natural, we hold that no serious theory of questions and answers should restrict itself to a treatment of redundant answers alone, and that it should be able to handle both."

H-Z then proceeds to develop a theory to account for all of these possible answers, by extending the PTQ grammar and the logic IL. This theory replaces the model theory of IL by what they call a "context-model." In essence this model is an IL-model expanded to include as model-theoretic objects the entire language of IL itself. Minimal answers are then translated into expressions that denote formulas when interpreted within the context of a preceding question. This is accomplished technically by including in the logic a set of context variables, and by including an abstraction over a context variable in the translation of the non-redundant answers. A context variable denotes an expression of IL, viz., the question that has set up the context. This idea of a context allows H-Z to define a semantics not just for questions like "Who dates Mary?", but also for each of the answers (a) through (e) in such a way that each of them is equivalent in extension.

H-Z's concern with the semantics of the answers to the questions, which at first sight seems to be our concern, is in fact another issue. For our theory, while it takes the answers of questions into account, is essentially not a theory of answers but a theory of questions. Of course, in the context of a more complete and user-friendly question-answering system, the ability to keep track formally of the context of the discourse and to express the answer in a number of different ways, is very attractive. Such a system would need the ability to go "backwards" from expressions in the logic to expressions in English with the same interpretation; Friedman (1981) discusses this issue from the point of view of the PTQ fragment. But the development in H-Z of the semantics of the questions themselves, although motivated from this different concern with the equivalence of redundant and non-redundant answers, does also lead them to an analysis of question semantics outside of the single semantics framework. Their analysis "lets questions denote different types of sets according to the type of that expression which is the critical one in any kind of answer." In other words, their semantic analysis of answers is quite similar to our pragmatic analysis of questions. The following table comparing the types assigned to various kinds of questions by their semantics and our pragmatics might help to make this analogy clearer.

Question class	Our typing	H-Z typing
yes-no	t	$\langle\langle s, \langle\langle s, t \rangle, t \rangle \rangle, t \rangle$
1 individual	$\langle e, t \rangle$	$\langle\langle s, f(T) \rangle, t \rangle$
2 individual	$\langle e, \langle e, t \rangle \rangle$	$\langle\langle s, f(T) \rangle, \langle\langle s, f(T) \rangle, t \rangle \rangle$

3.6.4 SCHA AND GUNJI

The work of Scha (1983) on the PHLIQA1 project and Gunji (1981), both being developed concurrently with the development of QE-III (Clifford 1982b), are remarkably similar in spirit, though not in detail, to the present work. The close parallels in the motivation of these three works indicate a trend among many researchers toward developing a formal foundation for computer systems that do natural language processing.

The major theoretical difference between QE-III and that of the PHLIQA1 project of Scha are that we make a distinction between the semantics and pragmatics of sentences in QE-III, so that the pragmatic interpretation of questions in QE-III is closely analogous to Scha's semantics for the same question. We continue to believe that this separation between the *denotation* of a sentence (given by the semantic component of the language) and its *interpretation* (given by the pragmatic component) is a simpler and more easily extendible approach to the problem of providing a formal account of meaning.

Much of the motivation for the work reported in Gunji (1981), namely to provide a formal pragmatics to a language specification by means of the computational application of a "superinterpreter" after the completion of the syntactic and semantic interpretation, is the same as ours. Gunji's superinterpreter, in fact, is quite clearly the computational realization of what we have termed our pragmatic interpretation. The major difference between these two projects is in the scope of their languages, which reflect Gunji's focus on conversation implicatures and ours on querying historical databases. Whereas Gunji's work covers declarative and imperative sentences, and true-false questions, QE-III resulted from a concentration on wh-questions and an explicit treatment of time-denoting expressions.

This concludes our informal discussion of the goals and philosophy behind the definition of the fragment QE-III, and its relation to other recent work in the area of formal question semantics. The next section provides an overview of QE-III through a series of examples and discussions illustrating the major features of the language. (The complete definition of the syntax, semantics, and pragmatics of QE-III can be found in Clifford (1987).)

4 EXAMPLES FROM THE QE-III FRAGMENT

4.1 INTRODUCTION

This section presents and discusses examples of the syntactic and translation rules of the QE-III fragment. As we pointed out in Section 3, the PTQ fragment

stands essentially intact as the core of QE-III. There are, however, certain changes to this core. One major change is our use of the logic IL_s as the intermediate translation language; this logic is a modification to Montague's IL, and makes explicit the "hidden" abstraction over indices that is a part of the evaluation process in Montague's PTQ analysis. With respect to IL, the major change is that in IL_s , we evaluate any expression α with respect to a state s by forming the expression $[\lambda i\alpha](s)$.

The other major aspect of QE-III is the inclusion of a formal pragmatic component to the language, on an equal standing with the syntactic and semantic components. The formalized pragmatic component of QE-III was primarily motivated by the desire to simplify the provision of an interpretation for questions in a formal way. As we showed in Section 3.5.6 the pragmatic interpretation of any sentential expression was essentially the denotation of the expression formed by λ abstracting over all of the free individual variables and also evaluating with respect to "now."

In addition to these changes in the underlying logic and method of evaluation, the following additional modifications have been made to the rules of the PTQ fragment.

1. Rule S4 has been modified to perform the single function of combining a term with an IV to form a sort of protosentence. It no longer performs the verb inflection for third-person singular present tense. The entire treatment of tense and time adverbials is now performed more systematically by rules S101–S106. (The tensing functions of S17 have therefore been totally eliminated.)
2. Montague's use of the variables he_0 and him_0 amounted to a simple technique of case marking in order to choose the appropriate personal pronoun upon substitution of a term. We have expanded this technique somewhat, using variables of the form [it-CASE- i] where CASE ranges over {NOM,DAT,ACC} and i over the natural numbers.
3. Rule S9 for combining a sentence adverbial ("Necessarily") with a sentence, has been eliminated. This is because the only sentence adverbials in QE-III are time adverbials, which are brought in together with the tense marker in rules S104–S106.
4. It is well known that there are problems with the PTQ treatment of conjunction and disjunction of terms and IVs (see discussion in Friedman (1979) and Bennett (1974)). While Friedman's bracketing solution is ultimately more acceptable (both by virtue of its generality and, of particular interest, its natural correspondence to a LISP implementation), we have for simplicity of presentation adopted Bennett's simple solution of marking all basic verbs with a # marker, which is removed when the verb is ultimately tensed. (We choose

this solution because the points we wish to make have only to do with the verbs, and are easily understood with this technique.

For ease of understanding the translations to follow, the following table shows the types of the variables used.

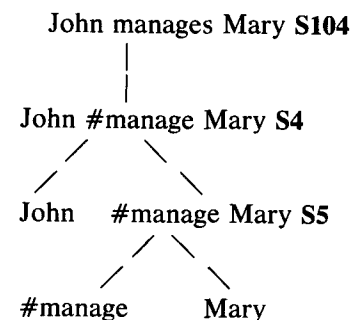
Variable symbols	Type of variable symbol
x, y, z, x_0, x_1, \dots	$\langle s, e \rangle$: individual concepts (ICs)
P, Q, Q_1, Q_2, \dots	$\langle s, \langle \langle s, e \rangle, t \rangle \rangle$: properties of ICs
p, q, q_1, q_2, \dots	$\langle s, t \rangle$: propositions
i	s : distinguished state variable <i>wrt</i> , which all expressions are evaluated
i_1, i_2, \dots	s : states
W	$\langle s, \langle \langle s, \langle \langle s, e \rangle, t \rangle \rangle, t \rangle \rangle$: properties of properties of ICs

4.2 PTQ-LIKE EXAMPLES FROM THE QE-III FRAGMENT

Before illustrating some of the added features of the QE-III database query fragment, we present a simple example within the syntactic range of the PTQ fragment (up to vocabulary differences) in order to contrast the way these two fragments derive and translate it. For example, under one analysis,

4-1. John manages Mary.

would have the following derivation tree in QE-III:



The syntactic and translation rules illustrated in this example are S4 to form an untensed clause from a SUBJECT and PREDICATE, S5 to form a verb phrase from a transverb and a direct object, and S104 to form a present-tense clause.

Several points arise with this example. First, we note that this analysis tree presents the derivational history of non-basic expressions in the language in the obvious way. Each node is labeled with a meaningful expression in QE-III; in case the expression is non-basic, it is further labeled by the syntactic rule by which it was constructed, and is given children labeled with the expressions from which it was obtained. Montague (1970b) provides a more formal definition of analysis trees; it should be sufficient to point out that the language is defined in such a way that to each analysis tree (though not necessarily to each meaningful expression) there corresponds a unique translation into the intermediate logical language.

This analysis of Example 4-1 illustrates several departures from the corresponding PTQ analysis. First, we note that the basic verb is prefixed with #, and this

prefix remains even after S4 is applied to combine the term "John" with the intransitive verb phrase "#manage Mary". Second, Rule S104 is new. It takes an untensed sentence as input and gives a (present) tensed sentence as output. Thus we have characterized tense as a property not of verbs, but of clauses, although this property in English is realized by the inflection of the main verb of the clause. The importance of this characterization will be made clearer when we consider the interaction of tense with interrogative sentences.

This method of introducing tenses into a sentence obviates the need for undoing the English verb inflections that would be required by a method (such as in PTQ or in Dowty (1979) that *always* introduced present tense first, subject to possible subsequent modifications. Dowty (1979) makes a similar point—though still in terms of introducing the tense via a SUBJ + PRED rule—but does not incorporate the idea into the fragment presented there.

In a number of the PTQ rules Montague makes use of the auxiliary notions of the gender of a CN or a T, and the third-person singular form of a verb. These notions are never defined with the same rigor that Montague demanded of other characteristics of his logic and grammar, presumably because he felt they were obvious and uninteresting. As in Bennett (1974) we make use of a number of similar auxiliary notions in our rules. This example points out two such notions, viz., that of the tense of a clause and the case of a variable. In our fragment a clause is either untensed or tensed, and belongs to a different category (though of the same logical type) in either case. A variable introduced into a sentence is either uncased, or one of NOM, ACC, or DAT.

The translation of Example 4-1 corresponding to the above analysis tree is as follows:

Mary ==> $\lambda P \exists x [P(i)(x) \wedge x(i) = \text{Mary}]$
 #manage ==> $\lambda W \lambda x [W(i)(\lambda i \lambda y [AS-1(y(i), x) \wedge EMP_*'(i)(y(i)) \wedge MGR'(i)(x)])]$
 #manage Mary ==> $\lambda W \lambda x [W(i)(\lambda i \lambda y [AS-1(y(i), x) \wedge EMP_*'(i)(y(i)) \wedge MGR'(i)(x)])](\lambda i \lambda P \exists x [P(i)(x) \wedge x(i) = \text{Mary}])$
 $\rightarrow \lambda x \exists z [AS-1(z(i), x) \wedge EMP_*'(i)(z(i)) \wedge MGR'(i)(x) \wedge z(i) = \text{Mary}]$
 John ==> $\lambda P \exists y [P(i)(y) \wedge y(i) = \text{John}]$
 John #manage Mary ==> $\lambda P \exists y [P(i)(y) \wedge y(i) = \text{John}]$
 $(\lambda i \lambda x \exists z [AS-1(z(i), x) \wedge EMP_*'(i)(z(i)) \wedge MGR'(i)(x) \wedge z(i) = \text{Mary}])$
 $\rightarrow \exists y [AS-1(\text{Mary}, y) \wedge EMP_*'(i)(\text{Mary}) \wedge MGR'(i)(y) \wedge y(i) = \text{John}]$
 John manages Mary ==> $\exists y [EMP_*'(i)(\text{Mary}) \wedge MGR'(i)(y) \wedge y(i) = \text{John} \wedge AS-1(\text{Mary}, y)]$

The pragmatic interpretation is represented by: $\exists y [EMP_*'(\text{now})(\text{Mary}) \wedge MGR'(\text{now})(y) \wedge y(\text{now}) = \text{John} \wedge AS-1(\text{Mary}, y)]$.

Our treatment of proper terms is slightly different

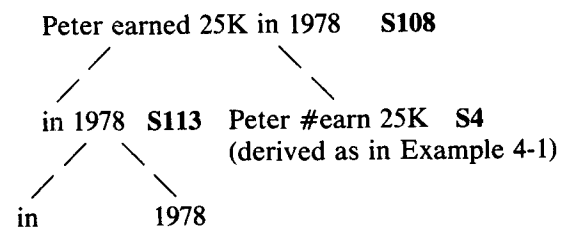
from the PTQ treatment, in that the translations include an individual concept variable whose extension at the state i is asserted to be the indicated individual. This is done because in HRDM all individuals of interest must be playing a role in the database, and roles can only be filled by individual concepts. Further, as we discussed in Section 3, verbs are treated as objects of the same type as in PTQ, but they are analyzed in terms of the database schema.

4.3 TEMPORAL REFERENCE IN QE-III

In addition to its indication by means of the tense system, temporal reference in English is also indicated by certain time adverbials (today, last year, . . .) and also by prepositional phrases (in 1978, on Monday . . .). Care must be taken in order to analyze properly the semantics of sentences that involve an interaction between tenses and these other temporal indicators. They cannot be applied sequentially as operators to a clause, or the semantics will be incorrect. (Dowty (1979) makes the same observation.) The following derivation for

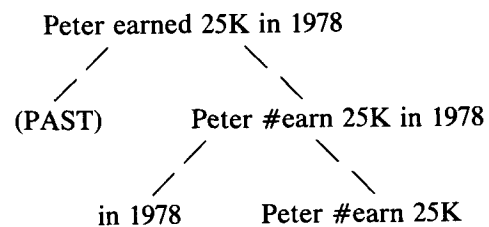
4-2. Peter earned 25K in 1978.

illustrates this aspect of QE-III.

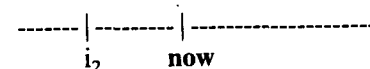


This example illustrates Rule S108, which simultaneously adds a tense (past) and a time adverbial, and S113, which forms a temporal prepositional phrase. The pragmatic interpretation correctly indicates that there is some state in the past that is also in the set of states 1978 at which the present-tense sentence "Peter earns 25K is true": $\exists i_1 \exists y [1978'(i_1) \wedge [i_1 < \text{now}] \wedge EMP_*'(i_1)(\text{Peter}) \wedge SAL'(i_1)(y) \wedge y(i_1) = 25K \wedge AS-1(\text{Peter}, y)]$.

If we had introduced the two temporal indicators (the tense and "in 1978") separately, in either order, the resulting interpretations would be incorrect.

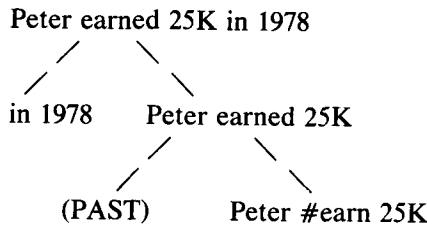


interpreted as: $\exists i_2 \exists i_1 \exists y [[i_2 < \text{now}] \wedge 1978'(i_1) \wedge EMP_*'(i_1)(\text{Peter}) \wedge SAL'(i_1)(y) \wedge y(i_1) = 25K \wedge AS-1(\text{Peter}, y)]$. This places the three times i_1 , i_2 , and **now** on the time line as follows:

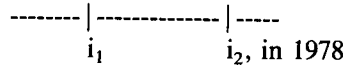


with i_1 anywhere on the time line in 1978.

The reverse order of sequential introduction is also incorrect.



since it is interpreted as: $\exists i_2 \exists i_1 \exists y [1978'(i_2) \wedge [i_1 < i_2] \wedge EMP_*(i_1)(Peter) \wedge SAL'(i_1)(y) \wedge y(i_1) = 25K \wedge AS-1(Peter, y)]$. Here the two times are located as follows:



The properties of Peter are asserted to be true in state i_1 , but i_1 may or may not be in 1978, and may or may not be in the past (with respect to **now**.) Only the simultaneous introduction of these temporal operators provides the correct interpretation.

Example 4-3 illustrates how tense is treated as a property of clauses in compound sentences, and how these tenses are independent of one another. It also illustrates how relative clauses are maintained in the QE-III fragment.

4-3. Peter manages an employee such that he earned 30K.

Under the most likely analysis, this sentence is interpreted in QE-III as: $\exists w \exists x \exists y \exists i_1 [EMP_*(\mathbf{now})(x(\mathbf{now})) \wedge MGR'(\mathbf{now})(w) \wedge w(\mathbf{now}) = Peter \wedge AS-1(x(\mathbf{now}), w) \wedge EMP_*(i_1)(x(\mathbf{now})) \wedge SAL'(i_1)(y) \wedge y(i_1) = 30K \wedge 1978'(i_1) \wedge [i_1 < \mathbf{now}] \wedge AS-1(x(\mathbf{now}), w)]$.

Example 4-4 illustrates how propositions can be treated in almost the same way as time constants for denoting sets of states.

4-4. John worked before Mary worked.

This sentence is analyzed as asserting that there was some state S_1 before **now** at which John worked, and that S_1 was also before some other state S_2 before **now** at which Mary worked.

$$\exists i_1 [[i_1 << (\lambda i_2 EMP_*(i_2)(Mary))] \wedge [i_2 < \mathbf{now}] \wedge [i_1 < \mathbf{now}] \wedge EMP_*(i_1)(John)].$$

Similarly, we can combine simple time expressions with prepositions to form temporal adverbials, as in Example 4-5.

4-5. Rachel worked before yesterday.

which is analyzed as: $\exists i_1 [[i_1 << \text{yesterday}'(\mathbf{now})] \wedge [i_1 < \mathbf{now}] \wedge EMP_*(i_1)(Rachel)]$. Notice that this translation places two restrictions upon when the state i_1 can occur in time.

1. $[i_1 << \text{yesterday}'(\mathbf{now})]$ because of "before yesterday," and
2. $[i_1 < \mathbf{now}]$ because of the past tense. Since a time before yesterday must be before **now** (by the meaning of "yesterday"), a meaning postulate for words such as "yesterday" might well be in order

here to remove this redundancy and reduce the final translation to $\exists i_1 [[i_1 << \text{yesterday}'(\mathbf{now})] \wedge EMP_*(i_1)(Rachel)]$.

We now proceed to discuss the other additional rules of the QE-III fragment. These rules either form expressions that have particular relevance to the database realm (possessives, role specifications, etc.) or form interrogative sentences. We will look first at the questions; some of the considerations involved in the framing of these rules for database querying purposes was given in Section 3.

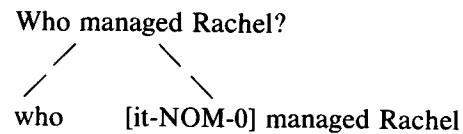
4.4 QUESTIONS IN QE-III

Consider the following query:

4-6. Who managed Rachel?

translated as $\exists i_1 \exists y [[i_1 < i] \wedge EMP_*(i_1)(Rachel) \wedge MGR'(i_1)(y) \wedge y(i_1) = u \wedge AS-1(Rachel, y)]$. Recall that the pragmatics provides a representation for the answer to questions, and that the pragmatic interpretation of this query is denoted by the expression $\lambda u \exists i_1 \exists y [[i_1 < \mathbf{now}] \wedge EMP_*(i_1)(Rachel) \wedge MGR'(i_1)(y) \wedge y(i_1) = u \wedge AS-1(Rachel, y)]$ formed by binding all free occurrences of the variable i to the constant **now**, and λ -abstracting over all of the free individual variables.

This example illustrates why the tense must be considered a property of the entire clause, rather than just of the verb phrase, if the semantics of the question is to come out right. For suppose instead that we derived (4-6) as follows:



The translation would then be $\exists y \exists i_1 [y(i) = u \wedge [i_1 < i] \wedge EMP_*(i_1)(Rachel) \wedge MGR'(i_1)(y) \wedge AS-1(Rachel, y)]$. The problem with this translation is that the manager-IC y is not tensed properly. When evaluated, this query will return the set of individuals u who are the extension of Rachel's manager-IC, not at some time in the past, but **now**. Because "who" has wider scope in this derivation, the past-tense operator could not capture the free i of the translation of "who". The question, under our treatment, is correctly analyzed as "Who (past) managed (past) Rachel?" rather than as "Who (**now**) managed (past) Rachel?" In order to get this reading, tenses (and tenses + TmADverbials) must be brought in last over all clauses, including interrogative sentences.

Interrogative terms (WHT's) can also be derived from common nouns and the interrogative determiners such as "which," as seen in Example 4-7.

4-7. Who manages which employees?

which is interpreted as: $\lambda u_1 \lambda u_2 \exists y [EMP_*(\mathbf{now})(u_2) \wedge MGR'(\mathbf{now})(y) \wedge y(\mathbf{now}) = u_1 \wedge AS-1(u_2, y)]$.

Example 4-8 illustrates a three-term interrogative, using the three-place verb "#supply" and a rule (a simple extension of the two-place case, essentially

taken from Dowty (1979) for combining such a verb with an indirect object to form a two-place verb.

4-8. What does who supply to whom?

The interpretation⁷ is $\lambda u_3 \lambda u_2 \lambda u_1 [\text{COMP}_* (\text{now})(u_3) \wedge \text{DEPT}_* (\text{now})(u_2) \wedge \text{ITEM}_* (\text{now})(u_1) \wedge \text{REL-3}(u_3, u_2, u_1)]$.

Example 4-9 illustrates a more complicated question that requires, in terms of the database representation, a "join" of two relations.

4-9. Who works for a department such that it sells shoes?

It is interpreted as $\lambda u \exists x [\text{EMP}_* (\text{now})(u) \wedge \text{DEPT}_* (\text{now})(x(\text{now})) \wedge \text{AS-1}(u, x) \wedge \text{ITEM}_* (\text{now})(\text{Shoes}) \wedge \text{REL-2}(x(\text{now}), \text{Shoes})]$.

Yes-no questions can take two forms in the fragments

4-10. Is it the case that Peter earns 30K?

and

4-11. Does Peter earn 30K?

Both of these questions receive the same interpretation: $\exists x [\text{EMP}_* (i)(\text{Peter}) \wedge \text{SAL}'(i)(x) \wedge x(i) = 30K \wedge \text{AS-1}(\text{Peter}, x)]$.

When-questions, very important in a historical database context, are illustrated by the following example.

4-12. When did Peter earn 25K?

interpreted as $\lambda i_1 \exists y [[i_1 < \text{now}] \wedge \text{EMP}_* (i_1)(\text{Peter}) \wedge \text{SAL}'(i_1)(y) \wedge y(i_1) = 25K \wedge \text{AS-1}(\text{Peter}, y)]$.

Finally, the next two examples illustrate the interaction of when and an already-formed term question, and the interaction of when with time phrases.

4-13. When did who manage whom?

interpreted as $\lambda u_2 \lambda u_1 \lambda i_1 \exists x [[i_1 < \text{now}] \wedge \text{EMP}_* (i_1)(u_1) \wedge \text{MGR}'(i_1)(x) \wedge x(i_1) = u_2 \wedge \text{AS-1}(u_1, x)]$, and

4-14. When and to whom did company A sell item B yesterday?

interpreted as: $\lambda i_1 \lambda u_2 \lambda u_1 \exists x [[i_1 < \text{now}] \wedge \text{yesterday}'(i_1) \wedge \text{DEPT}_* (i_1)(u_1) \wedge x(i_1) = u_1 \wedge \text{COMP}_* (i_1)(A) \wedge \text{ITEM}_* (i_1)(B) \wedge \text{REL-3}(A, B, u_1)]$.

This concludes the examples of the kinds of queries expressible in the language QE-III, and the semantics and pragmatics that the fragment provides for them. We now present some of the other additions we have made to the PTQ fragment in order to express certain other common query constructions.

4.5 MISCELLANEOUS FEATURES OF QE-III

The use of possessives is very common in database queries, and is easily incorporated into the fragment, as in

4-15. Who is Peter's manager?

which is interpreted as $\lambda u \exists x [\text{MGR}'(\text{now})(x) \wedge x(\text{now}) = u \wedge \text{AS-1}(\text{Peter}, x)]$. An alternative way of phrasing the same question uses "of" instead of the possessive marker.

4-16. Who is a manager of Peter?

and ultimately receives the same interpretation. Finally, specification of the role played by an individual in the

database can also be accomplished by means of the word "as".

4-17. Who has Peter as manager?

interpreted as $\lambda u \exists z [\text{MGR}'(\text{now})(z) \wedge z(\text{now}) = \text{Peter} \wedge \text{AS-1}(u, z)]$, or by the simple concatenation of the role and a term.

4-18. Who sells Item 37?

interpreted as $\lambda u [\text{DEPT}_* (\text{now})(u) \wedge \text{ITEM}_* (\text{now})(37) \wedge \text{REL-2}(u, 37)]$.

5 CONCLUSION

The problem of modeling the semantics of time is one which is beginning to be explored by researchers in a number of different areas of computer science. We believe that formal logic can make an important contribution to our understanding and specification of the properties of time that we can incorporate into our models and systems. Using the logic IL_s and the framework of MS, we have presented in this paper an overview of the HRDM, which is a formalization of the concept of a historical database. HRDM provides for the modeling of historical information in a DBMS, for the specification of constraints on the way that information can change over time, and for a query language for accessing that information with specific reference to its temporal dimension.

To complement the relational query language of HRDM (Clifford and Croker 1987), we have in this paper described a formal English database query language, QE-III, which is defined in a MS framework. QE-III incorporates a formal syntax, semantics, and pragmatics to account for an interpretation of questions that accord with the interpretation of HRDM, including an account of multiple-wh questions, a semantics and pragmatics of time, and a grammar that is conducive to a computer implementation. In addition to its formal syntax and parallel semantics, QE-III is provided with a formal pragmatics that provides a representation for the answer(s) to a question as a function of its syntax and semantics. We believe that this approach, and the whole area of formal pragmatics as a component of language theory, is a fertile area for further research.

ACKNOWLEDGEMENTS

This paper is based upon research done while I was a graduate student in computer science at SUNY Stony Brook. I would like to thank my advisor David Warren for introducing me to, and schooling me in, the complexities of Montague Semantics, and also Professor Joyce Friedman for encouraging me to conduct this research. I would also like to thank the referees for their many useful and constructive comments on an earlier draft of this paper.

REFERENCES

- Ariav, G.; Beller, A.; and Morgan, H.L. 1984 *A Temporal Model* (Technical Report DS-WP 82-12-05). Department of Decision Sciences University of Pennsylvania, Philadelphia, PA.
Bar-Hillel, Yehoshua 1954 Indexical Expressions. *Mind* 63: 359-379.

- Belnap, Nuel D., Jr. 1982 Questions and Answers in Montague Grammar. In Peters, S., and Saarinen, E. (eds.) *Processes, Beliefs, and Questions*. D. Reidel Publishing Co., Dordrecht, The Netherlands.
- Ben-Zvi, J. 1982 *The Time Relational Model*. Ph.D. diss., Department of Computer Science, University of California, Los Angeles, CA.
- Bennett, M.R. 1974 *Some Extensions of a Montague Fragment of English*. Ph.D. diss., UCLA.
- Bennett, M.R. 1977 A Response to Karttunen on Questions. *Linguistics and Philosophy* 1: 279–300.
- Bennett, M.R. 1979 *Questions in Montague Grammar* (Technical Report). Indiana University Linguistics Club, Bloomington, IN.
- Bennett, M.R. and Partee, B.H. 1972 *Toward the Logic of Tense and Aspect in English* (Technical Report). Indiana University Linguistics Club, Bloomington, IN.
- Carnap, R. 1947 *Meaning and Necessity*. University of Chicago Press, Chicago, IL.
- Chang, C.L. 1978 DEDUCE 2: Further Investigations of Deduction in Relational Data Bases. In Gallaire, H. and Minker, J. (eds.) *Logic and Data Bases*. Plenum Press, New York, NY.
- Clifford, J. 1982 A Model for Historical Data Bases. In *Proceedings of Logical Bases for Data Bases*. ONERA-CERT, Toulouse, France.
- Clifford, J. 1982 *A Logical Framework for the Temporal Semantics and Natural-Language Querying of Historical Databases*. Ph.D. diss., Department of Computer Science, SUNY at Stony Brook.
- Clifford, J. 1985 Towards an Algebra of Historical Relational Data bases. In *ACM-SIGMOD International Conference on Management of Data*. Austin, TX.
- Clifford, J. 1987 *Natural Language Querying of Historical Data bases—The QE-III Language Definition and Examples* (Technical Reports CRIS-#164, GBA-WP#87-90). Department of Information Systems, New York University, New York, NY.
- Clifford, J., and Croker, A. 1987 The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans. In *Proceedings of the Third International Conference on Data Engineering*. Los Angeles, CA.
- Clifford, J., and Warren, D.S. 1983 Formal Semantics for Time in Data bases. *ACM Transcript on Data base Systems* 8(2): 214–254.
- Codd, E.F. 1970 A Relational Model of Data For Large Shared Data Banks. *Communications of the ACM* 13(6): 377–387.
- Cooper, R. 1979 Variable Binding and Relative Clauses. In Guenther, F., and Schmidt, S.J. (eds.) *Formal Semantics and Pragmatics for Natural Languages*. D. Reidel Publishing Co., Dordrecht, The Netherlands.
- Dowty, D.R. 1978 *A Guide to Montague's PTQ* (Technical Report). Indiana University Linguistics Club, Bloomington, IN.
- Dowty, D.R. 1979 *Word Meaning and Montague Grammar*. D. Reidel Publishing Co., Dordrecht, The Netherlands.
- Engdahl, E. 1986 *Constituent Questions—The Syntax and Semantics of Questions with Special Reference to Swedish*. D. Reidel Publishing Co. Dordrecht, The Netherlands.
- Filmore, C. 1968 The Case for Case. In Bach, E. and Harris, R. (eds.) *Universals in Linguistic Theory*. Rinehart & Winston, New York, NY.
- Friedman, J. 1979 An Unlabeled Bracketing Solution to the Problem of Conjoined Phrases in Montague's PTQ. *Journal of Philosophical Logic*, 8:151–169.
- Friedman, J. 1981 Expressing Logical Formulas in Natural Language. In Groenendijk, J.; Janssen, T.; and Stokhof, M. (eds.) *Formal Methods in the Study of Language*. Mathematical Center, Amsterdam, Holland.
- Friedman, J. and Warren, D.S. 1978 A Parsing Method for Montague Grammars. *Linguistics and Philosophy*, 347–372.
- Gadia, S.K. and Vaishnav, J. 1985 A Query Language for a Homogeneous Temporal Data Base. In *Proceedings of The Fourth Annual ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*.
- Gallaire, H. and Minder, J. 1978 *Logic and Data Bases*. Plenum Press, New York, NY.
- Green, B.F.; Wolf, A.K.; Chomsky, C.; and Laughery, K. 1963 Baseball: An Automatic Question Answer. In Feigenbaum, E.A., and Feldman, J. (eds.) *Computers and Thought*. McGraw-Hill, New York, NY.
- Groenendijk, J. and Stokhof, M. 1983 On the Semantics of Questions and the Pragmatics of Answers.
- Gunji, Takao 1981 *Toward a Computational Theory of Pragmatics—Discourse, Presupposition, and Implicature*. Ph.D. diss., The Ohio State University, Columbus, OH.
- Hamblin, C.L. 1973 Questions in Montague English. *Foundations of Language* 10: 41–53.
- Harris, L.R. 1973 *The ROBOT System: Natural Language Processing Applied to Database Query* (Technical Report TR78-1). Department of Mathematics, Dartmouth College, Hanover, NH.
- Hasbrouck, B.L. 1982 *Methods of Parsing English with Context-Free Grammar*. Master's thesis, Department of Computer Science, SUNY at Stony Brook.
- Hausser, R. and Zaefferer, D. 1978 Questions and Answers in a Context-Dependent Montague Grammar. In Guenther F., and Schmidt, S.J. (eds.) *Formal Semantics and Pragmatics for Natural Languages*. D. Reidel Publishing Co., Dordrecht, The Netherlands.
- Hendrix, G.G.; Sacerdoti, E.D. and Sagalowicz, D.; and Slocum, J. 1978 Developing a Natural Language Interface to Complex Data. *ACM Transactions on Database Systems* 3(2).
- Hintikka, J. 1974 Questions About Questions. In Munitz, M.K. and Unger, P.K. (eds.) *Semantics and Philosophy*. New York University Press, New York, NY.
- Karttunen, L. 1977 Syntax and Semantics of Questions. *Linguistics and Philosophy* 1: 3–44.
- Kasher, A. 1977 What is a Theory of Use. *Journal of Pragmatics* 1: 105–120.
- Kuno, S. and Robinson, J. 1973 Multiple WH-Questions. *Linguistic Inquiry* 3(2): 463–487.
- Landsbergen, S.P.J. 1981 Adaptation of Montague Grammar to the Requirements of Parsing. In *Proceedings of the Third Amsterdam Colloquium on Formal Methods in the Study of Language*. Mathematical Centre, Amsterdam, Holland.
- Levinson, Stephen C. 1983 *Pragmatics*. Cambridge University Press, Cambridge, England.
- Lum, V., et al. 1984 Designing DBMS Support for the Temporal Dimension. In *ACM-SIGMOD International Conference on Management of Data*. Boston, MA.
- Maier, D. 1983 *The Theory of Relational Data Bases*. Computer Science Press, Rockville, MD.
- Marciszewski, W. 1971 Introduction to Appendix. In Pelc, Gerzy (ed.) *Studies in Functional Logical Semiotics of Natural Language*. Mouton, The Hague.
- Martin, R.M. 1959 *Toward a Systematic Pragmatics*. North-Holland, Amsterdam.
- Montague, R. 1968 Pragmatics. In Klibansky, R. (ed.) *Contemporary Philosophy: A Survey*. Florence: La Nuova Italia Editrice, Florence, Italy.
- Montague, R. 1970 Pragmatics and Intensional Logic. *Synthese*, 22: 68–94.
- Montague, R. 1970 English as a Formal Language. In Visentini, B. et al (eds.) *Linguaggi nella Societa e nella Tecnica*. Edizioni di Comunita, Milan, Italy.
- Montague, R. 1973 The Proper Treatment of Quantification in English. In Hintikka, K.J.J. et al. (eds.) *Approaches to Natural Language*. D. Reidel Publishing Co., Dordrecht, The Netherlands.
- Montague, R. and Thomason, R. (ed.) 1974 *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven, CT.

- Morris, C. 1938 Foundations of the Theory of Signs. In *International Encyclopedia of Unified Science*. University of Chicago Press, Chicago, IL.
- Morris, C. 1946 *Signs, Language, and Behavior*. Prentice Hall, New York, NY.
- Pope, E.N. 1976 Questions and Answers in English. In *Janua Linguarum Series Practica* 226. Mouton, The Hague.
- Quirk, R. and Greenbaum, S. 1974 *A University Grammar of English*. Longman, London, England.
- Reiter, R. 1978 On Closed World Data Bases. In Gallaire, H., and Minker, J. (eds.) *Logic and Data Bases*. Plenum Press, New York, NY.
- Rescher, N. and Urquhart, A. 1971 *Temporal Logic*. Springer Verlag, New York, NY.
- Roberts, P. 1954 *Understanding Grammar*. Harper & Row, New York, NY.
- Scha, Remko J.H. 1983 *Logical Foundations for Question Answering*. Ph.D. diss., University of Groningen, The Netherlands.
- Schank, R.C. 1972 Conceptual Dependence: A Theory of Natural Language Understanding. *Cognitive Psychology* 3(4): 552-631.
- Snodgrass, R. 1984 The Temporal Query Language TQuel. In *Proceedings of the Third ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*. Waterloo, Ontario, Canada.
- Snodgrass, R., and Ahn, I. 1985 A Taxonomy of Time in Data Bases. In *ACM-SIGMOD International Conference on Management of Data*. Austin, TX.
- Stalnaker. 1972 Pragmatics. In Davidson, Donald and Harman, Gilbert (eds.) *Semantics of Natural Languages*. D. Reidel Publishing Co., Dordrecht, The Netherlands.
- Tichy, P. 1978 Questions, Answers, and Logic. *American Philosophical Quarterly* 15(4): 275-284.
- Vendler, Z. 1967 *Linguistics in Philosophy*. Cornell University Press, Ithaca, NY.
- Waltz, D.L. 1978 An English Language Question Answering System for a Large Relational Data base. *Communications of the ACM* 21(7): 526-539.
- Warren, D.S. 1979 *Syntax and Semantics in Parsing: An Application to Montague Grammar*. Ph.D. diss., University of Michigan, Ann Arbor, MI.
- Woods, W.A.; Kaplan, R.M.; and Nash-Webber, B. 1972 *The LUNAR Sciences Natural Language Information System: Final Report* (Technical Report BBN Report 2378) Bolt, Beranek and Newman Inc., Cambridge, MA.

NOTES

1. See Clifford (1987) for the definition of IL_s, and a discussion of how (and why) it differs from Montague's IL.
2. In all translation examples we follow Partee in using a double arrow "= \Rightarrow " to indicate the immediate result of applying a translation rule of the fragment, and a single arrow " \rightarrow " to indicate the result of any of a number of logical simplifications (principally λ -reduction).
3. The "such that" construction for heading relative clauses is a syntactic holdover from the PTQ fragment, which, because our concern is primarily semantic, we have not attempted to replace with a more sophisticated treatment. For a treatment of more usual English relative clauses, the reader is referred to Cooper (1979).
4. Although David Warren suggests considering the following sort of exchange:
[SHE]: "There are several Fire Island ferries each day."
[HE]: "Oh, really! When do they arrive and when do they leave?"
It is natural to interpret this as a request for the set of ordered pairs $\langle t_1, t_2 \rangle$ representing the arrival and departure times of particular ferry runs. Other sorts of multiple when-questions that ask for a range ("Between when and when . . .") seem to be of this same type.
5. Tichy (1978) makes many of the same points that we make here regarding the proposition idea.
6. Actually their syntax does not cover direct questions, and so this is really their treatment of "John loves him" in the category of Basic Question; it seems clear, however, that they intend the semantics of the corresponding direct question to be the same.
7. REL-3 indicates that there is a tri-ary relationship among the indicated three individuals.