AI-TR-483

DETERMINING THE SCOPE OF ENGLISH QUANTIFIERS

KURT A. VANLEHN

JUNE 1978

ARTIFICIAL INTELLIGENCE LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AI-TR-483	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Determining the Scope of English Quantifiers		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Kurt A. VanLehn		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0643
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd Arlington, Virginia 22209		12. REPORT DATE June 1978
		13. NUMBER OF PAGES 127
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Quantification	Meaning Representation
Natural Language Understanding	Semantic Rules
Quantifier Scope	Anaphora Rules
Semantic Interpretation	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

How can one represent the meaning of English sentences in a formal logical notation such that the translation of English into this logical form is simple and general? This report answers this question for a particular kind of meaning, namely quantifier scope, and for a particular part of the translation, namely the syntactic influence on the translation.

Rules are presented which predict, for example, that the sentence

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract, con't.

Everyone in this room speaks at least two languages.

has the quantifier scope $\forall\exists$ in standard predicate calculus, while the sentence

At least two languages are spoken by everyone in this room.

has the quantifier scope $\exists\forall$.

Three different logical forms are presented, and their translation rules are examined. One of the logical forms is predicate calculus. The translation rules for it were developed by Robert May (May 1977). The other two logical forms are Skolem form and a simple computer programming language. The translation rules for these two logical forms are new.

All three sets of translation rules are shown to be general, in the sense that the same rules express the constraints that syntax imposes on certain other linguistic phenomena. For example, the rules that constrain the translation into Skolem form are shown to constrain definite np anaphora as well.

A large body of carefully collected data is presented, and used to assess the empirical accuracy of each of the theories.

None of the three theories is vastly superior to the others. However, the report concludes by suggesting that a combination of the two newer theories would have the greatest generality and the highest empirical accuracy.

This report describes research done while a National Science Foundation Fellow at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Project Agency of the Department of Defense under Office of naval Research contract N00014-75-C-0643.

Determining the Scope of

English Quantifiers

by

Kurt A. VanLehn

Massachusetts Institute Institute of Technology

June 1978

Revised version of a dissertation submitted to the Department of Electrical Engineering and Computer Science on January 20, 1978 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

How can one represent the meaning of English sentences in a formal logical notation such that the translation of English into this logical form is simple and general? This report answers this question for a particular kind of meaning, namely quantifier scope, and for a particular part of the translation, namely the syntactic influence on the translation.

Rules are presented which predict, for example, that the sentence

Everyone in this room speaks at least two languages.

has the quantifier scope $\forall\exists$ in standard predicate calculus, while the sentence

At least two languages are spoken by everyone in this room.

has the quantifier scope $\exists\forall$.

Three different logical forms are presented, and their translation rules are examined. One of the logical forms is predicate calculus. The translation rules for it were developed by Robert May (May 1977). The other two logical forms are Skolem form and a simple computer programming language. The translation rules for these two logical forms are new.

All three sets of translation rules are shown to be general, in the sense that the same rules express the constraints that syntax imposes on certain other linguistic phenomena. For example, the rules that constrain the translation into Skolem form are shown to constrain definite np anaphora as well.

A large body of carefully collected data is presented, and used to assess the empirical accuracy of each of the theories.

None of the three theories is vastly superior to the others. However, the report concludes by suggesting that a combination of the two newer theories would have the greatest generality and the highest empirical accuracy.

Acknowledgements

I would like to take this opportunity to thank those who have lent me their support in this enterprise. First and foremost, I would like to thank my thesis advisor, Jon Allen, for his consel over the years. I am deeply indebted to him for reading the first, almost incomprehensible, report on this research.

I would particularly like to thank my fellow students -- Candy Sidner, Mitch Marcus, Dave McDonald, Beth Levin and Brian Smith -- for their criticism and advice. If I hadn't had their support, I could never have attempted this research. Special thanks go to Brian, Mitch and Beth for proofreading.

I am indebted to my informants. Their friendly cooperation when faced with the exasperating task of casting quantifier scope judgements always amazed me and heartened me.

Lastly, I thank my roomates, and especially my friends Vicki Bier, Ilene Horvitz and Don Coen, for their constant encouragement. I cannot express how much their support has meant to me. I am deeply grateful for it.

CONTENTS

FIGURES

1. Introduction

The original motivation for the research reported here was to improve the performance of a natural language understanding system, LUNAR (Woods et. al. 1972). The component of LUNAR that disambiguated the scope of quantifiers seemed to make too many mistakes. It was thought that by merely importing some recent research in transformational linguistics, namely Kroch 1974, the disambiguation algorithm could be improved.

However, Kroch's theory was unclear in a few points. While collecting data to clarify Kroch's work, it soon became apparent that people usually do not disambiguate quantifier scope. This suggested that quantifier scope correlations, such as those predicted by LUNAR's rules or Kroch's rules, are epiphenomena. That is, they appear to be a side effect of some other linguistic phenomena, or the result of a degraded version of some real linguistic process.

Since then, the research has concentrated on an accurate description of these correlations. It was hoped that this would uncover the linguistic process that was causing the correlations, and eventually lead to an improvement in LUNAR's disambiguation algorithm. However, even after a huge corpus was collected -- well over 1500 judgements were collected and hundreds of pages of natural text were analyzed -- the situation is inconclusive.

Nonetheless, the correlations are much clearer now, and three clear candidates have emerged as possible underlying processes for quantifier scope correlations. Improvement of LUNAR, however, seems remote.

1.1 The Quantifier Scope Problem

The classic example of the quantifier scope problem, which first appeared in Chomsky 1957, is the active/passive pair

(1a) Everyone in this room speaks at least two languages.
(1b) At least two languages are spoken by everyone in this room.

Although these two sentence have the same "lexical content", they have different syntactic structures and different meanings. It is traditional to give (a) the reading where each person may speak a different two languages, and (b) the reading where the same two languages are spoken by everyone.

If one were to represent these two readings in predicate calculus, they would differ only in the scopes of the quantifiers:

(2a) $\forall x$ [(x is in this room) $\supset$ [$\exists y$ (y is two languages) & (x speaks y)]]

(2b) $\exists y$ [(y is two languages) & [$\forall x$ (x is in this room) $\supset$ (x speaks y)]]

In (a), the existential quantifier is inside the scope of the universal quantifier. Thus (a) could be true in a room where everyone spoke different languages. (b) would be false in that room, since the existential quantifier is outside the scope of the universal quantifier. (b) would only be true in a room where everyone speaks the same two languages. Note that the predicates and their arguments are the same in both expressions. Thus, the two sentences of (1) have the same lexical content.

The quantifier scope problem is just this: why do (1a) and (1b) have different meanings even though they have the same open class words (i.e. nouns, verbs, adjectives, and adverbs) and the same predicate/argument relations? The quantifier scope problem is not to delineate all the factors which give these sentences their meanings, for some of those factors involve discourse context and pragmatic knowledge, and there are as yet no adequate formalizations of such influences.[1]

---

1. My favorite example of the influence of pragmatics is a play on Chomsky's example:

(4a) Everyone at PARC uses a dialect of LISP.
(4b) Everyone at IJCAI uses a dialect of LISP.

Most people in the AI community know that the Palo Alto Research Center (PARC) maintains the programming language INTERLISP, thus they probably use it exclusively. So (a) has the interpretation that they all use the same dialect of LISP. In predicate calculus, the quantifier order would be $\exists\forall$. But at IJCAI, the biannual conference for the field, one finds people from all over the world. Since there are many versions of LISP in use, (b) must mean that the conference attendees are using different versions of LISP -- the $\forall\exists$ order in predicate calculus.

Consequently, the problem is restricted to finding just the syntactic influence. In collecting data, one should control for lexical content, which I take to be the choice of open class words and predicate/argument relations. An excellent review of the quantifier scope problem can be found in Ioup 1975.

Sometimes the quantifier scope problem is taken to include problems with negation, modality, conditionals, conjunctions, or the quantificational adverbs (eg. ***often***). This paper investigates only relationships between noun phrases. Also, the many problems associated with the article ***any*** will be ignored.

1.2 Ducking the Really Hard Subproblem

The criteria for evaluating solutions to the quantifier scope problem are the usual ones: empirical adequacy and theoretical economy. That is, the predictions of the theory ought to match the trends in the data, and secondly, the framework and possibly even the rules that operate inside that framework ought to be shared with theories of other linguistic phenomena. However, there is one aspect of the data that makes the joint satisfaction of these two criteria exceedingly difficult.

The relative strengths of the lexical and syntactic influences is significantly different for quantifier scope than for other linguistic phenomena. Lexical content is much more important in quantifier scope judgements than in, say, the acceptability of np movements or definite np anaphora.[1] As an example, take the clauseboundedness constraint.

It is well known that certain np movements, such as passive, dative and complex np shift, are limited to the clause containing them (for simplicity, I'm ignoring np raising). Similarly, reflexive pronoun anaphora requires antecedents to be in the clause containing the reflexive pronoun. Thus, the (a) sentences below are acceptable, but the (b) sentences are not.

---

1. Throughout this report, some standard linguistic terminology will be employed. "np" is short for "noun phrase", "pp" for "prepositional phrase". Two constituents are "clausemates" if they are members of the same clause.

(5a) John blurted out that the beer was laced with LSD.
(5b) * The beer was blurted out that someone laced with LSD by John.

(6a) People believe that John killed himself.
(6b) * John believes that people killed himself.

In general, quantifier scope is also clausebound. That is, if an existentially quantified np is to be inside the scope of a universal np, then the existential np must be in the same clause as the universal np. For example,

(7a) John blurted out that each senator was offered a TV set.
(7b) A TV set blurted out that each senator was offended.

sentence (a) has the reading "a different TV set per senator was offered" since the existential quantifier over TV sets can be inside the universal quantifier over senators. In (b), there can be only one TV set, since the existential must be outside the universal.

Thus, it seems that quantifier scope, np movement, and reflexive pronoun anaphora are all clausebound. However, it is not difficult to use lexical content to override the clauseboundesness of quantifier scope. An example is

(8) A quick test confirmed that each drug was psychoactive.
[4] a different test per drug
[2] all the drugs were involved in a single test
[4] ambiguous between the previous two readings
[1] one test with many parts
[1] a coordinated battery of tests

The numbers in square brackets preceding each reading is the number of informants that got that reading. The first reading, where ***a quick test*** is inside the scope of ***each drug***, violates clauseboundedness since the existential np is not in the clause that the universal np, ***each drug***, is in. On the other hand, it is very difficult to violate the clauseboundedness of np movement or reflexive pronoun anaphora. Indeed, I know of no counterexamples.

The weakness of the clauseboundedness correlation is typical of the other quantifier scope correlations. Sentences can be constructed whose lexical content is strong enough to violate almost any syntactic rule one could write. On the other hand, most linguistic phenomena studied to date are more highly constrained by syntax. So to be empirically adequate, a theory of quantifier scope must sacrifice its similarity to other linguistic phenomena.

The only research to recognize this problem was done by Robert May (May 1977). He asserts that his rules generate the "unmarked" readings. Counterexamples to the rules are "marked" interpretations -- they should be less frequent. The marked/unmarked distinction has occasionally appeared in linguistics, especially in phonology. At this point in time, however, I believe it is fair to say that markedness is not at all well understood. In particular, there is no way to explain why the marked interpretations of quantifier scope occur more frequently than the marked constructions of syntax.

I choose to duck the problem in a different way. I will assume that quantifier scope correlations are epiphenomena. That is, I assume that certain phenomena correspond to syntactically real processes. These actually use the syntax of a sentence to perform their task -- eg. disambiguation of predicate/argument relations, or coreference relations. However, there is no such process for quantifier scope. Instead, the informant must "misuse" one of the real processes to disambiguate quantifier scope, perhaps with the aid of a general cognitive mechanism for performing analogy. It seems plausible that when a real process performs a task that it is not suited for, nor often used for, it would break down under strong lexical pressure. Thus, postulating that quantifier scope correlations are epiphenomena explains, in a sloppy intuitive way, why syntax has a weaker influence over quantifier scope than it has over constituent movements and anaphora.

The idea that quantifier scope isn't a real process also explains certain difficulties of data collection. Every informant has, at one time or another, asked to be excused from making a judgment. When a sentence is constructed so that syntax doesn't immediately affirm the reading that lexical content would lead one to prefer, then people appear to think very hard before casting their judgments. Quite often, they would read the sentence through, paraphrase it back, and yet be unable to answer the kinds of questions that would illuminate their quantifier scope judgements -- they would reread the sentence several times before answering such questions. This seems to indicate that they were doing quantifier scope disambiguation after they had understood the sentence in the usual way. Although these observations are informal and subjective, they lend some plausibility to the suggestion that quantifier scope is disambiguated by "misusing" syntactically real processes, rather than the disambiguation being an integral part of the understanding of every sentence.

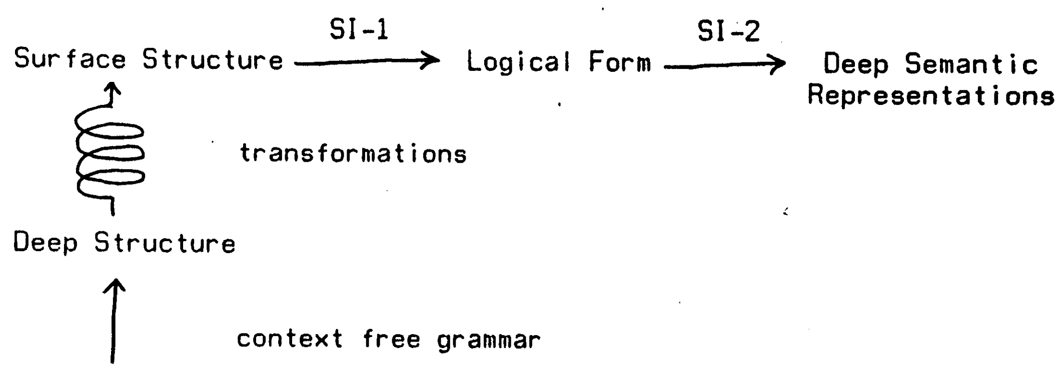
1.3 Overview

The idea that scope is epiphenomenal immediately raises the question of which real process is being misused. That question is the organizing theme of this report. Three theories are presented that purport to solve the quantifier scope problem. They are based on three linguistic phenomena: transformations, anaphora, and lexical composition. These theories are preceded by a section giving a descriptive account of the data.

The transformational theory was developed by Robert May (May 1977). It will be reviewed in some detail since it is, in many ways, the best theory of quantifier scope to date. The other two theories are original, although the basic ideas of the anaphoric theory have appeared in the works of many linguists, notably Keenan 1974 and Reinhart 1976.

All three theories are compatible with the view held by the lexical-interpretive school of linguistics. This view can be illustrated with a diagram:

(9)



In this view, "meaning" is derived from the surface structure directly rather than via the deep structure.

There is currently a controversy concerning how predicate/argument relations should be represented in surface structure. Chomsky and his students use "traces" that point from the argument position to the constituent that fills the argument (see section 3.1). For example, the object position in a passive sentence would have an invisible trace that points to the subject. Bresnan and her students use lexical procedures to "undo" constituent movements, such as passive and raising. The theories presented below are, for the most part, neutral with respect to this controversy. Quantifier scope judgements appear to depend on the actual location of a constituent in surface structure, and not on the position of its trace, if it has been moved.

Each of the three theories proposes a particular logical form, and a particular SI-1 map. (I have found it convenient to relabel the latter with the less cumbersome name "translation", since the map translates the surface structure into logical form.) The transformational theory's logical form is a version of the typed predicate calculus. The anaphoric theory uses typed Skolem form. The lexical theory's logical form is similar to programming languages, such as LISP or ALGOL.

The variation in logical forms forces an interesting extension of the usual linguistic methodology. Earlier works have taken the logical form to be pretheoretically given. In fact, all logical forms I have seen are versions of the typed predicate calculus.[1] This report considers the design of the logical form to be an integral part of a theory of quantifier scope. That is, each theory claims that its logical form is correct.

The criteria for judging logical form are taken to be quite different from those for judging deep structure, the most famous of the two remote structures. In the lexical-interpretive theory of grammar, deep structure and transformations work together as a sort of syntactic well-formedness checker.[2] A sentence is well-formed if and only if there exists a legal deep structure and a legal transformational derivation of the sentence from that deep structure. The deep structure has little to do with the meaning of sentences. It is just a repository for certain syntactic generalizations -- eg. the X-bar convention and SVO ordering.

---

1. Jackendoff 1972 is an exception. His Modal Structure appears to be isomorphic to Skolem form.
2. I am indebted to Mitch Marcus for this insight.

Thus, for a lexical-interpretive theorist, there is just one criterion for judging the design of deep structure: does the design facilitate elegant expression of the syntactic well-formedness constraints of natural language?

Logical form, on the other hand, is supposed to be a representative of the meaning of sentences. Woods suggests the following four criteria for evaluating a logical form (adapted from Woods 1978, page 17):

(10a) It must be precise, formal and unambiguous.

(10b) It must be capable of representing any interpretation that a human reader can place on a sentence.

(10c) It should facilitate subsequent intelligent processing of the resulting interpretation.

(10d) It should facilitate an algorithmic translation from English sentences into their corresponding semantic representations.

Predicate calculus does a respectable job of meeting criteria (a), (b) and (c). Its formality, precision and lack of ambiguity can be demonstrated by giving it a formal semantics; that is, by devising an algorithm that, given an expression of predicate calculus and a model of the world, calculates whether the the expression is true in that model. The world model associates a set of objects with each undefined term (i.e. provides the extension of the term). Criterion (b), namely the expressive adequacy of predicate calculus, can be tested only by experience. Suffice it to say that predicate calculus would not be so widely used today, a century after its invention, if there were numerous sentences that it could not represent. Criterion (c) is can be met by predicate calculus by writing formal rules of inference. Given an expression, such rules can, in principle at least, draw conclusions that one would call "intelligent".

This report concentrates on criterion (d). By proper design of the logical form, the translation rules can be made very simple. Moreover, the rules can be made theoretically economical, in the sense that they apply, for example, to both anaphora and quantifier scope. In the anaphoric and lexical theories, a great deal of theoretical economy is gained by proper design of the logical forms.

However, criterion (a) has not been entirely ignored. An appendix has been provided that informally demonstrates that each of the logical forms introduced has a formal semantics. In addition to dispersing any doubts about formality and precision, the appendix is meant to clarify the reader's (and the author's!) intuitive understanding of the logical forms' meanings.

It is difficult to challenge a venerable logical form such as predicate calculus on the basis of its expressive adequacy -- criterion (b). Indeed, there are just two empirical arguments in the quantifier scope literature that claim that predicate calculus is not expressively adequate. One is presented in Jackendoff 1972. It is based on the famous sentences

(11a) I told many of the men three of the stories.
(11b) I told three of the stories to many of the men.

Jackendoff notes that there are three distinct quantifier scope interpretations of the two sentences, but only two quantifiers. Since predicate calculus represents quantifier scope by operator order, it can represent only two interpretations. This argument is successfully refuted in Fauconnier 1975 by adding the collective indefinite quantifier to predicate calculus. The other argument, from Hintikka 1974, is refuted in section 4.3. A new expressive adequacy argument, which is presented in section 5.2, could also be refuted by adding a new operator to predicate calculus. I expect that this is a general pattern. It is probably always possible to patch up the expressive inadequacies of predicate calculus.

No attempt has been made to provide rules of inference for these logical forms. It is possible therefore that they may fail to meet criterion (c). Indeed, Woods claims that one of the logical forms, Skolem form, has just this flaw (Woods 1975). In particular, he claims that inference rules concerning negation are intractable.

It should be pointed out that this report judges logical forms only on their facility for representing quantifier scope intuitions. In particular, the predicate/argument notion, which has recently come under attack for its imprecision (Smith 1978) and its empirical inadequacy (Levin, in preparation), is used freely in the logical forms below. When only quantifier scope intuitions are considered, the predicate/argument notion turns out to be adequate and theoretically convenient.

In short, the evaluation of logical forms will be based on the simplicity of the quantifier scope translation rules. The basic ideas behind the three sets of rules are, it turns out, somewhat similar.

The transformational theory of quantifier scope was developed by Robert May (May 1977). It is based on rules from the revised extended standard theory of transformations, or "trace theory" as it is more commonly called (Chomsky 1976). The basic device is a rule, QR, which moves quantified nps out of their surface structure position, and attaches them just above an S node. The movement leaves behind a trace, which is bound to the moved np. That is, the movement puts a bound variable where the np occurred, and puts a quantifier to bind it at the front of some clause.

The movement is constrained by two rules, Subjacency and the Condition on Proper Binding. Subjacency forces the quantifier to be attached to the smallest clause which contains the bound variable. Thus, in

(12a) Some woman said every senator was sick.
(12b) ∃x:woman() [(x said [∀y:senator() (y was sick)])]
(12c) ∀y:senator() [∃x:woman() [(x said (y was sick))]]

sentence (a) has reading (b) and not (c). The Condition on Proper Binding is a well formedness condition on logical form. It forces a bound variable to be inside the scope of the quantifier that binds it. Hence, in

(13a) Some woman in every city voted democrat.
(13b) ∀x:city() [∃y:woman-in(x) (y voted democrat)]]
(13c) * ∃y:woman-in(x) [∀x:city() (y voted democrat)]]

(a) must have logical form (b) since (c) is ill-formed. These two constraints are well motivated, since they are used to constrain transformations (i.e. the map from deep to surface structure).

The anaphoric theory is a combination of the work of Edward Keenan (Keenan 1974) and Tanya Reinhart (Reinhart 1976). It's basic idea is that the ∀x∃y reading is markedly different from the ∃y∀x reading. The ∀x∃y read is indicated in logical form by providing the type function of y with an extra argument which is filled by x. For example,

(14a) Ron talked to each woman about a problem.
(14b) { ∃y:problem(x), ∀x:woman(), (Ron talked to x about y) }
(14c) { ∃y:problem(), ∀x:woman(), (Ron talked to x about y) }

sentence (a) has both (b) and (c) as readings. Logical form (b) represents Ron's talking about a different problem to each woman, and (c) represents his talking about the same problem to all of them. Note that the left-to-right ordering of the quantifiers no longer matters, since the function/argument relation represents the quantifier scope. This is the basic idea of Skolem form -- to represent quantifier scope explicitly, with the function/argument relation.

The linkage of the two nps via the function/argument relation is constrained by the same rules that constrain definite pronoun coreference. That is, an np with a universal quantifier must "c-command" the existentially quantified np in order to be allowed to link to it. "X c-commands Y" means roughly that X is higher than Y in the syntax tree. Hence, in

(15a) Every mathemetician speaks a foreign language.
(15b) { ∃y:foreign-language(x), ∀x:mathemetician(), (x speaks y) }
(15c) { ∃y:foreign-language(), ∀x:mathemetician(), (x speaks y) }
(15d) A foreign language is spoken by every mathemetician.

sentence (a) can have (b) or (c) as an interpretation, but (d) can have only (c), because ***every mathemetician*** doesn't c-command ***a foreign language*** in (d) while it does in (a).

The typed Skolem form used in the anaphoric theory is also subject to a well formedness constraint, namely that a function may not depend on itself for an argument. Thus,

(16a) Every woman in an eastern city voted democrat.
(16b) { ∀x:woman-in(y), ∃y:city(), (x voted democrat) }
(16c) * { ∀x:woman-in(y), ∃y:city(x), (x voted democrat) }

the only well formed interpretation of (a) is (b). In (c), woman-in depends on y, which depend indirectly on x. So woman-in depends on itself, and the expression is ill-formed.

Lastly, the typed Skolem form's formal semantics is designed so that a dummy functional argument can not be distinguished from an argument supplied by a np in surface structure. Thus,

(17a) Some candidate in each election is corrupt.
(17b) { ∀x:election(), ∃y:candidate(x), (y is corrupt) }
(17c) Some candidate is corrupt in each election.
(17d) { ∀x:election(), ∃y:candidate(), (y is corrupt) }

sentence (a) is unambiguous since it must have the interpretation (b). That is, since ***each election*** is inescapably an argument of ***some candidate***, the sentence must be interpreted to imply that a different candidate per election is corrupt. However, sentence (c) can have either (b) or (d) as a reading. If it has the (b) reading, then it too will imply that a different candidate per election is corrupt. Crucially, they is no way to distinguish, for the sake of quantifier scope correlations anyway, whether x is a dummy argument of candidate as in (c) or a lexically realized argument of candidate, as in (a).

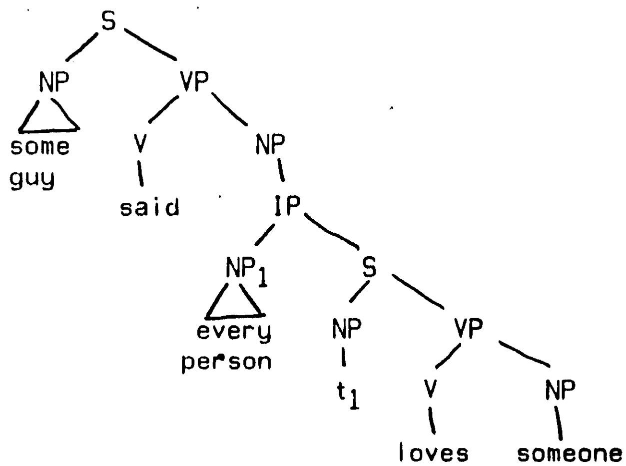
All the constraints, except the last one, are obeyed by definite np coreference. So the anaphoric theory has good independent motivation.

The lexical theory is based on a very common, very important phenomenon. Unfortunately, very little is known about this phenomenon, so the independent motivation of the theory is weaker than the other two. Lexical composition is the process which builds the word-meaning of a constituent from the word-meaning of other constituents. This process is widely held to be constrained by Strict Compositionality -- the lexical content of a constituent is built from the lexical content of its daughters, not its sisters or some other constituents in the the syntax tree. In natural language engineering, this constraint means one need only pass semantic markers up, not over or down.

The logical form for the lexical theory is like a computer programming language in that it has a "for loop" operator, called an "iteration phrase". The basic idea is that universal nps are the loop variables of iteration phrases. The ∀x∃y reading is represented by an existential np y which is inside the iteration phrase that x is the loop variable of. Hence, when (a) has the logical form (b)

(18a) Some guy said that every person loves someone.

(18b)



it means some particular guy claimed that every person loves a different person, since ***someone*** is inside the iteration phrase, but ***some guy*** is not.

The translation into this logical form is just like semantic marker passing. The iteration phrase is passed up the tree, starting from the universal np. Like QR, this movement leaves behind a trace bound to the moved np.

One of the constraints on this movement, which is motivated by an observation of Vendler's (Vendler 1967), is that the iteration phrase (henceforth, IP) must end up dominating a predicate which is worth iterating. Vendler noted that "Take each apple" sounds odd. But "Weigh each apple" sounds fine. ***Weigh*** has two distinct interpretations -- weighing each apple individually, or weighing the whole basketful of apples at once. On the other hand, ***take*** doesn't have two such distinct readings, so the iteration is pointless. Hence, when the IP dominates ***weigh***, the sentence is fine, but when the it dominates ***take***, there isn't a predicate worth iterating, so the sentence sounds odd. Thus, Vendler's observation motivates one constraint on IP raising, as the lexical theory is called.

The other constraints are, unfortunately, unmotivated. First, the clauseboundedness of quantifier scope is captured by stipulating that it cost "effort" to raise an IP. The cost is proportional to the number of nodes the IP must rise through. The second stipulation involves the formal semantics of the logical form. Basically, sentences such as

(19) Every flight to an eastern city was late.

where ***an eastern city*** is not iterated (i.e. all the flights go to the same city) are accounted for by stipulating that the logical form be evaluated (i.e. its extension calculated) in argument order. Hence, since ***flights to an eastern city*** is an argument of the IP, it is evaluated before the iteration takes effect. Hence, ***an eastern city*** is not part of the iteration. These two constraints are completely unmotivated, but they do predict the quantifier scope correlations.

The most empirically accurate of the three theories is the lexical one. But the best motivated one is the anaphoric theory. The only thing that prevents their combination is a lack of certain crucial anaphoric data. The last section details this problem.

However, none of the three theories predicts the data with an accuracy that demands conviction. This could be due to incorrect theories. However, it is my belief that the mismatch is due to the epiphenomenal nature of quantifier scope. People do not <u>do</u> quantifier scope.

2. *A Description of the Major Correlations*

This section presents a descriptive account of the correlation of quantifier scope judgments and syntactic structure. The account is divided into three parts. The first concerns the influence that articles have on quantifier scope. The other two parts concern the positions of nps in syntactic structure. Part two describes how embedding an np in various constructions influences its quantifier's scope. The third part discusses the influence that left-to-right ordering has on nps at the same level of embedding (eg. clausemate nps).

The syntactic structures discussed are always surface structures, not deep structures. Thus, for example, by the "object" of a passive sentence, I will mean the np appearing directly after the verb, not the superficial subject.

The intuitions of the informants will be described using two informal relations: the different/per relation and the same/per relation. I have found this presentation much less confusing than one based on predicate calculus. These two relations will be defined by example. Consider this ambiguous sentence and its two interpretations.

(20) Ron talk to each woman about a problem.

(20a) a different problem per woman
(20b) the same problem per woman

(21a) $\forall$x:woman [$\exists$y:problem [Ron talked to x about y]]
(21b) $\exists$y:problem [$\forall$x:woman [Ron talked to x about y]]

The (a) interpretation will be called the different/per reading, and (b) will be called the same/per reading. It is convenient to consider these readings to be binary relations between nps, ie.

(23) When "the same NP1 per NP2" or "a different NP1 per NP2"
call NP1 the "subject" of the per relation, and
call NP2 the "object" of the per relation.

Thus one says "the np ***each woman*** is the object of the per relation of either interpretation, and ***a problem*** is the subject." This nomenclature makes many correlations much easier to describe.

2.1 Correlations with Articles

Articles are a very important factor in the translation from surface structure to logical form. It seems, moreover, that the effect of the object np's article on its per relation is independent of the effect of the subject np's article.

In particular, the object of the different/per relation often bears the article ***each***, while ***a(n)*** is often the article of the subject of a different/per relation. These observations are supported by statistical data gathered from a large corpus of expository text. Since data from this corpus appears throughout this section, it is worth a moment to discuss its preparation.

The text came from technical papers written by people in the MIT AI laboratory. Several corpora were used, of about 2000 sentences each. The text was filtered to remove sentences which could display neither per relation. Sentences with just one np were removed. Assuming that objects of per relations must be plural, sentences which lacked plural nps were removed. More controversially, it was assumed that a number marking on the subject of the per relation is necessary in order to get an unequivocal judgment. So sentences which had neither singular nps nor nps with numeric modifiers (eg. ***three***, ***several***, ***a few***) were eliminated from consideration. In one corpus, for example, this filtering left 121 np pairs to examine more closely.

After the text was filtered, the sentences were read carefully, and the np pairs were assigned one of the two per relations. When I found it difficult to judge which relationship an np pair had, I would look the sentence up in its context. If that failed to disambiguate the interpretation, I would consult that sentence's author.[1] Thus, the readings are "forced" intuitions, in the sense mentioned in the introduction.

Figure 1 shows the distribution of per relations over the articles. The effect of surface structure has been, hopefully, washed out -- the only constraint on the

1. On one occasion, the author intended the sentence to be ambiguous. The $\forall\exists$ reading was most appropriate to the immediate context of the sentence, but the $\exists\forall$ reading was in fact true as well. The idea that ambiguity is sometimes desirable challenges some deeply rooted beliefs. In particular, an extreme version of this idea is that the quantifier scope problem is not a problem; instead, our models of inference have a problem in that they prefer unambiguous expressions.

Fig. 1. Correlations of the Per Relations with the Articles

The rous are the articles of the object np.
The columns are the articles of the subject nps.
The numerators are the number of different/per readings.
The denominators are the sum of different/per and same/per readings.

	a	the	one	some	several	total
each	6/6	2/3	0/0	1/1	0/0	9/10
every	2/3	1/1	0/0	0/0	0/0	3/4
all	1/4	0/3	0/1	0/0	0/1	1/9
any	1/2	0/2	0/0	0/0	0/0	1/4
plural	9/60	1/27	2/5	0/2	0/0	12/94
total	19/75	4/36	2/6	1/3	0/1	26/121

---

surface location of the nps was that they occured in the same sentence. The important points to notice are: 90% of the time, nps with ***each*** were the object of the different/per relation, not the same/per relation. 25% of the time, nps with ***a(n)*** were the subjects of the different/per relation. 100% of the np pairs that had ***each*** and ***a(n)*** as their articles had the different/per interpretation. And lastly, that 85% of the different/per readings had either an ***each*** on the object np, or an ***a(n)*** on the subject np. In short, among np pairs that can show a per relation, ***each*** and ***a(n)*** mark the different/per relation while their absence marks the same/per relation.

The above correlation may be a side effect of a correlation between the articles and the lexical content of sentences that determines quantifier scope judgments. It may also be due to a correlation between articles and the positions of the nps in surface structure. To determine the influence of the articles alone, groups of sentences were constructed which controlled for lexical content and surface structure. For example

(24a) The club president splashed each member with a glass of champagne.
(24b) The club president splashed many of the members with several glasses of champagne.
(24c) The club president splashed all the members with a glass of champagne.

Since the only difference between these sentences is in the articles of the two nps, any variation in quantifier scope intuition must be attributed to the variation of the articles. However, since the quantifier scope judgment is so subtle, the way such sentences are presented to the informant can have a large influence on the results. After some informal experimentation, the following "flashcard" mode of presentation was adopted.

Each sentence was typed on a file card, and submitted to the informant to be read silently. This avoided any contribution that intonation might make toward disambiguation of the sentence[1]. To avoid mental pollution from prior lexical contents, informants were only shown one sentence from any given paraphrase set. To avoid fatigue (and hostility!), informants were never asked to analyze more than five sentences at a time.

The judgements were ellicited somewhat indirectly. I would start by asking the informant to paraphrase the sentence. Often, this was enough to determine whether they were giving the sentence a different per reading, or a same per reading. If their reply was noncommittal, I would ask them questions, eg

Every guy kissed a girl.

(25a) Did they all kiss the same girl?
(25b) If there are 5 guys, how many girls does this imply got kissed?
(25c) Is there a different girl per guy?

Often, people would find these questions quite difficult to answer. Even after lengthy pondering, some people hadn't the slightest preference for one reading over the other. These judgements were counted as half different/per, half same/per in the total.

The results of such presentations are indicated below by appending to the front of each sentence the percentage of the informants who thought the sentence tended to have a different/per relation, rather than a same/per relation. For example, the results of the paraphrase group cited above, and another one very much like it, are:

---

1. Anthony Krock claims that an intonation break, such as a slight pause, prevents an np following the break from including an np preceding the break within its scope. See Krock 1974.

(26a) 80% The c.p. splashed each member with a glass of champagne.
(26b) 30% The c.p. splashed all the members with a glass of champagne.
(26c) 50% The c.p. splashed many of the members with several glasses of champagne.

(27a) 90% The c.p. splashed a glass of champagne over each member.
(27b) 0% The c.p. splashed a glass of champagne over all the members.
(27c) 0% The c.p. splashed several glasses of champagne over many of the members.

These percentages should not be taken too literally. The addition of another couple of judgments sometimes made the percentages swing up or down by 5 or 10 percentage points, but rarely by more than that.

The results of these two groups, and many others, support several generalizations. First, the articles of the objects of the per relations can be arranged in a hierarchy:

(28) each > every > all of the > all the > other plural articles

The higher an article on the hierarchy, the greater the likelihood that its np pair will have a different/per reading. This hierarchy has been seen before in the linguistics literature (eg. Ioup 1975). It is known to model the acceptability of nps when they fill certain arguments of certain "collective" predicates, such as ***meet***, ***swarm***, ***gather***, ***embrace***, etc. The following example illustrates how the hierarchy predicts the acceptability of various nps as the subject of ***meet***.

(29)
* Each man met.
*? Every man met.
?? All of the men met.
? All the men met.
The men met.

The explanation for this variation is based on two assumptions. First, an np can be interpreted either "collectively" or "distributively". Loosely speaking, the collective interpretation of an np yields a set, while the distributive interpretation yields a quantified variable, ranging over individuals. The articles influence whether an np will receive the collective or the distributive interpretation. In particular, the higher on the hierarchy an article is, the more its np tends to receive the distributive interpretation. In particular, ***each*** nps are always distributive, and ***the*** nps are almost always collective.

The second assumption is that certain predicates require certain of their arguments to be a set in order to make sense. For example, when ***meet*** is used intransitively, as in

(30a) * The man met at the pub.
(30b) The couple met at the pub.

(a) is unacceptable because it takes two or more men to make up a meeting. But (b) is acceptable, because it meets this selectional restriction.

If one assumes that a distributively interpreted np has the same semantic features, so to speak, as it would have if its article were the singular ***the***, then the varying acceptability of (29) is explained. Such an assumption also explains the following contrast.

(31a) * Each man met at the pub.
(31b) Each couple met at the pub.

(a) is bad because (30a) is bad. (b) is acceptable because (30b) is acceptable.

Now, to explain why the hierarchy also correlates with the different/per reading, one needs the following stipulation:

(32) If an np is the object of the different/per relation, then it must be interpreted distributively.

Thus, most different/per objects have ***each*** as their article because ***each*** most clearly marks the distributive interpretation. That the distributive/collective hierarchy is relevant to quantifier scope is thus the first observation one can make concerning the articles.

A second observation is that definite nps are usually the subjects of the same/per relation, rather than the different/per relation. The following example illustrates the point.

(33a) 100% Little Billy received a toy from each of his aunts.
(33b) 0% Little Billy received the toy from each of his aunts.

(a) has a clear different/per reading. However, the definite article ***the*** in (b) prevents this reading, resulting in a nonsensical same/per interpretation. This observation is not particularly surprising. However, there turns out to be a dialect that treats partitives (ie. nps of the form " <article> of <np>") as definite nps. In that dialect, partitives can not be the subject of the different/per relation.

The partitive dialect shows up in

(34) 50% Little Billy received one of the toys from each of his aunts.

All the informants were quick and sure of their judgments on this sentence. But half of them thought it was nonsense, and the other half thought it was a perfect sentence. There were no long, head scratching pauses, nor complaints that "it really doesn't say" or that "it depends on context", which usually accompanied the analysis of other ambiguous sentences. Moreover, the informants with same/per readings on this sentence tended to have same/per readings on other sentences involving partitives. That the judgments are rapid, and consistent across individuals, is evidence of a partitive dialect[1].

The partitive dialect has cropped up occasionally in the syntactic analysis of certain constructions[2] such as

(35a) There was a dealer at the party.
(35b) * There was the dealer at the party.
(35c) % There was one of the dealers at the party.

(36a) Speaking of the dealer, have you ever seen his car?
(36b) * Speaking of a dealer, have you ever seen his car?
(36c) % Speaking of one of the dealers, have you ever seen his car?

(37a) The book is John's.
(37b) * A book is John's.
(37c) % One of the books is John's.

(38a) Big as the demonstration was, the police maintained order.
(38b) * Big as a demonstration was, the police maintained order.
(38c) % Big as one of the demonstrations was, the police maintained order.

where "%" indicates a dialect split. Such examples motivate describing the partitive

---

1. I'd like to suggest that dialectal variations, such as the partitive dialect, is excellent evidence for the linguistic reality of the process that underlies the variation. Interestingly, in all the data I have collected on quantifier scope, I have observed only this dialect, and another dialect involving WH questions, which is presented in section 3. I have not found a dialectal preference for, say, the different/per reading, or surface ordering of quantifiers, or any rule related to quantifier scope alone. I suspect that further research will never uncover a true quantifier scope dialect.
2. See Stockwell 1973 page 118.

dialect with a feature that often crops up in the linguistics literature: specificity. The basic idea is that all definite nps are specific, and some indefinite nps are specific. Whether or not partitives are specific can vary across dialects, thus explaining the (c) judgements. For example, existential ***there*** is taken to require a nonspecific np as the object. Hence, (35a) is okay, (35b) is unacceptable, and (35c) is okay only in dialects where partitives are nonspecific.

The specific interpretation can be defined, in a loose sort of way, in terms of the presuppositions of np reference (Readers unfamiliar with the use of presuppositions in the linguistic literature may wish to skip this paragraph). The presuppositions of definite nps are separated into those that are unique to nps with true blue definite articles, such as *the*, *that*, *those*, etc, and those presuppositions that are shared by partitives as well. Let nps whose uniqueness and existence is presupposed, be said to receive the "specific" interpretation. Both definite nps and partitives would receive the specific interpretation. Other presuppositions, such as identifiability, would be reserved for true blue definite nps alone. Thus, one would describe the distribution of articles in the above syntactic environments by requiring the appropriate np to have a specific interpretation or, in the case of (35), a nonspecific one. The dialectal variations of the (c) sentences are easily explained by whether or not the informants give partitives a specific interpretation.

Although the notion of presupposition may not be a good way to think of the specific interpretation, the interpretation itself is just what is needed to describe the influence of certain articles on quantifier scope judgments. One simply replaces the original observation that definite nps can only be the subjects of same/per relations, with the following stipulation:

(39) If an np is the subject of a different/per relation, then it must receive the nonspecific interpretation.

In the partitive dialect, partitives are specific and hence can be only same/per subjects. On the other hand, nps with the article *a(n)* are almost always nonspecific. Hence they very frequently occur as the subjects of different/per relations.

Although it is tempting to form a specific/nonspecific hierarchy, I believe such a hierarchy would be far less accurate than the distributive/collective hierarchy[1]. Specificity depends too strongly on things other than the np's article. For example, the first np of a clause is almost always specific, regardless of its article. Indefinite nps with a great deal of descriptive content, such as long relative clause modifiers, tend to be specific. The adjectives ***certain*** and ***particular*** often bias their nps toward a specific interpretation. But despite its context dependence, the specific interpretation plays an important role in certain theories of logical form, as will be seen shortly.

To summarize, the influence of articles can be described with the aid of two binary distinctions, the collective/distributive interpretation and the specific/nonspecific interpretation. In order to have a different/per relation, the object must be distributive and the subject must be nonspecific. Otherwise, the np pair receives a same/per reading. Distributive interpretations are correlated with a simple hierarchy of articles:

(40) each > every > all of the > all the > other plural articles

The specific/nonspecific distinction can not be so simply described. However, ***a(n)*** is usually nonspecific, and definite nps are almost always specific.

2.2 Assumptions Regarding Specificity and Distributivity

In the previous section, specificity and distributivity were shown to be important correlates of quantifier scope intuitions. Since the theories to be presented make heavy use of these notions, this section has been provided to clarify them, and indicate their relationship to other kinds of np interpretations.

Throughout the rest of this report, it will be assumed that the three article features

---

1. Georgette Ioup (Ioup 1975) has proposed a hierarchy that combines the object and subject articles. Unfortunately, she was unable to place the indefinite singular articles, ***a(n)*** and ***some***, in her hierarchy. The preferences regarding the indefinite plural articles can probably be explained in terms of pragmatic content -- Ioup herself observes that the numerousity of the article affects quantifier scope preferences (ie. ***many*** is greater than ***a few***, so it has a stronger tendency to be involved in a same/per readings). In short, there little reason to believe that Ioup's hierarchy captures inherent variations in the specificity of indefinite articles.

(41) distributive / collective
specific / nonspecific
definite / indefinite

are independent, even though of the eight possible feature combinations, two are quite rare in English, namely definite/nonspecific/distributive and definite/nonspecific/collective. However, a significant theoretical economy is realized by considering all eight combinations to exist in principle.[1] Figure 2 has examples of all eight combinations.

It should be noted that some sort of pseudo-anaphoric modifier is necessary to create a nonspecific definite np. The reader may have noted the use of ***previous*** and ***associated*** in the figure. These modifiers bring the relationship between the two nps of the different/per relation perilously close to anaphora, intuitively. If the relationship is indeed one of anaphora, and not quantifier scope, then the argument that nonspecific definite nps exists breaks down.

In one corpus of natural text, five examples of definite nps as subjects of different/per relations occurred. Two used the adjectives ***previous*** and ***corresponding***. The other three nps, underlined, occured in

(42a) The packets associated with each active node are shown after the node description, followed by a slash.

(42b) For each sequence, that critical displacement for which the locally parallel pairings were just perceptible was determined.

(42c) At each point in the parsing process, the parser executes the action of the rule of highest priority whose pattern matches.

These sentences all have different/per readings, with the underlined nps as the subjects of the per relation. But parts of the nps' descriptions, especially in (a), seem to verge on coreferring with the ***each*** np. So the character of the internominal relationship -- anaphrora or quantifier scope -- is somewhat indeterminate. It seems difficult, therefore, to show that true blue nonspecific definite nps exist. On the

1. The eight combinations do not exhaust the number of ways to interpret nps. The generic interpretation, for example, isn't represented. Note also that the distributive/collective issue is moot when the np has a singular determiner.

Fig. 2. Eight Combinations

Spec	Distri	Def	Example
+	+	+	Each day, each girl kisses a boy. (same girls per day, different boy per girl)
+	+	-	Each day, certain girls try to catch a boy and kiss him. (same girls per day, different boy per girl)
+	-	+	Each year, a cruise to those ports makes an extraordinary profit. (same ports per year, same cruises per port)
+	-	-	Each year, a cruise to several particularly exotic ports makes an extraordinary profit. (same ports per year, same cruise per port)
-	+	+	For each node, the associated packets contain a packet mother that knows the name of the node. (different packets per node, different mother per packet)
-	+	-	Each day, many girls try to catch a boy and kiss him. (different girls per day, different boy per girl)
-	-	+	Each node is linked to the mother of the previous nodes. (different previous nodes per node, same previous nodes per mother)
-	-	-	Each day, a cruise to exotic foreign ports leaves Commonwealth Pier. (different ports per day, same cruise per port).

Using the following tests:

1. If NP2 is not a PP or possessive modifier of NP1, and their interpretation is that there is a different NP1 per NP2, then NP1 is nonspecific and NP2 is distributive. (see section 2.1)

2. If NP2 is a PP or possessive modifier of NP1, and their interpretation is that there is the same NP1 per NP2, then NP2 is collective. (see section 4.2)

3. If NP2 is a topicalized time adverb with the article ***each***, and NP1 is in the subject of the clause, and the two nps have the same/per interpretation, then NP1 is specific.

other hand, no one knows how to capture the subtle kinds of anaphora that link the modifiers of (b) and (c) to their *each* nps. So, it is not unreasonable to call their relationship "quantifier scope" and use nonspecific definite nps in our logical form to capture such dependencies.

It should be pointed out that specificity also figures in discussions of negation and opacity. However, it may turn out that the sort of specificity that conditions quantifier scope is different from the specificity that conditions negation and opacity. For example, if there are four distinct interpretations of

(43) Each sister wants to have a MIT prof over for supper.

namely,

(44a) They both want to dine with Jon, who is an MIT prof.

(44b) Connie wants to invite Jon to supper, and Ilene wants to invite Ira, who is also an MIT prof.

(44c) They both want just one MIT prof at the dinner party, but they don't care who.

(44d) Connie and Ilene each want to be allowed to invite a different MIT prof over, but they haven't decided which ones to invite yet.

then opacity and quantifier scope intuitions are independent. The four readings correspond to the four possible combinations of the two per interpretation with the transparent/opaque distinction. So in (c), for example, ***a MIT prof*** could be specific to quantifier scope, since (c) is the same/per reading, but nonspecific to opacity, since (c) is the opaque reading of *want*'s complement. The question is, do the syntactic features that correlate with quantifier scope specificity (eg. definiteness, length of descriptive content, surface grammatical role, etc.) also correlate with opacity judgements? If so, then there is only one kind of specificity. Since this question is as yet unanswered, one should allow the possibility that there may be two kinds of specificity, and take "specificity" in the sequel to refer only to the kind of specificity that is correlated with quantifier scope judgements.

2.3 Correlations with Embedding Constructions

Although less influential than articles, the relative "depth" of nps in surface structure also effects which per relation will be reported. All types of np-embedding constructions are, in a sense, related. In particular, embedding an np in a clause, and embedding it in a prepositional phrase, are just two ends of the same scale. This point is clearly demonstrated with groups of paraphrases, such as the one shown in figure 3.

In all the sentences, the subject of the per relation is a np which is modified by an embedding structure that contains the object of the per relation. The figure shows that when the embedding structure is a clause (ie. a full relative clause, abbreviated as FRC in the figure), the np pair uniformly receives a same/per interpretation. On the other end of the scale, where the embedding structure is a determiner (ie. a possessive np, "det" in the figure), the np pair always receives a different/per

---

Fig. 3. Correlations with the Form of NP Modifiers

With ***each*** embedded:

FRC:	0%	At the conference yesterday, I managed to talk to a guy who is representing each raw rubber producer in Brazil.
RRC:	50%	At the conference yesterday, I managed to talk to a guy representing each raw rubber producer in Brazil.
pp:	100%	At the conference yesterday, I managed to talk to a representative from each raw rubber producer in Brazil.
det:	100%	At the conference yesterday, I managed to talk to each raw rubber producer's representative.

With ***every*** embedded:

FRC:	0%	At the conference yesterday, I managed to talk to a guy who is representing every raw rubber producer in Brazil.
RRC:	0%	At the conference yesterday, I managed to talk to a guy representing every raw rubber producer in Brazil.
pp:	85%	At the conference yesterday, I managed to talk to a representative from every raw rubber producer in Brazil.
det:	100%	At the conference yesterday, I managed to talk to every raw rubber producer's representative.

reading. In the middle of the scale, where the embedding structure is a gerund or a prepositional phrase, the sentences are more ambiguous. The gerund embedding (ie. reduced relative clause, RRC in the figure), tends to have a same/per reading, while the prepositional phrase tends to have a different/per reading.

The four forms of np modification can be arranged in a hierarchy according to their tendency to occur with the different/per relation:

(45) determiner > pp > gerund > clause

This hierarchy, which will be henceforth be called the embedding hierarchy, can be seen in statistical data as well -- see figure 4. Unfortunately, embedding structures containing the appropriate articles occurred too sparsely to verify much of the hierarchy.

Figures 1 and 4 show that the influence of the embedding hierarchy is less than the influence of the distributive/collective hierarchy. Reducing an *each* to an *every* has more effect (delta = 36% for statistical data, 63% for paraphrastic data) than reducing a pp to a gerund (delta = 26% for statistical data, 47% for paraphrastic data). It would be interesting to construct a more extensive comparison of the two hierarchies.

Fig. 4. The Embedding Hierarchy in Statistical Data

	different/per readings	both per readings	percentage dif./per
Embedded ***each***			
FRC	0	1	0%
RRC	3	5	60%
pp	6	7	86%
Embedded ***every***			
FRC	0	0	-
RRC	0	1	0%
pp	2	4	50%
Embedded ***all***			
FRC	0	0	-
RRC	0	0	-
pp	3	14	21%

Embedding constructions that modify nps are only one kind of embedding construction. But the embedding hierarchy can be seen in other kinds of embedding constructions as well. Figure 5 presents a group of sentences that have various forms of nominalizations as their subjects. Again, a hierarchy is evident, with possessive np nominalizations at the different/per extreme, and full, clausal nominalizations at the same/per extreme. However, many of the sentences are barely acceptable as English sentences, making this data somewhat unconvincing. The unacceptability seems unrelated to the per relation, however, since the sentences still sound odd when ***the demonstrators*** is substituted for ***each demonstrator***. Further investigation is advisable before extending the embedding hierarchy to cover nominalizations.

2.4 The Asymmetry of Embedding

So far, all the embedding examples have embedded the object of the per relation, and placed the subject np outside the embedding construction. When these positions are reversed, a hierarchy is again evident:

Fig. 5. The Embedding Hierarchy and Subject Nominalizations

Lexical Nominalizations

100%	Each demonstrator's release required a short hearing.
100%	The release of each demonstrator required a short hearing.

Gerund Nominalizations

100%	Freeing each demonstrator required a short hearing.
100%	Each demonstrator's being released required a short hearing.
71%	The court's freeing each demonstrator required a short hearing.

Infinite Nominalizations

71%	To free each demonstrator would have required a short hearing.
72%	For each demonstrator to be released would have required a short hearing.
50%	For the court to free each demonstrator would have required a short hearing.

That-S Nominalization

*	That the court release each demonstrator would require a short hearing.

(46a)	66%	Striking airline workers forced several major airlines to cancel every flight which was going to an eastern airport today.
(46b)	25%	Striking airline workers forced several major airlines to cancel every flight going to an eastern airport today.
(46c)	0%	Striking airline workers forced several major airlines to cancel every flight to an eastern airport today.
(46d)	0%	Striking airline workers forced several major airlines to cancel an eastern airport's flights today.

Here, the object np ***an eastern airport*** has been embedded relative to the subject np ***every flight***. This hierarchy is the reverse of the one found when it was the object np that was embedded. In this hierarchy, the clausal embedding enhances the different/per interpretation instead of the same/per relation.

If the per relations are represented in predicate calculus, then it is easy to state a generalization that covers both hierarchies. Let Q be either the universal or existential quantifier, and let R be the other -- ie. the existential or universal quantifier, respectively. The correlation of quantifier scope readings and the embedding hierarchy can be stated as:

(47) Let X be the category of a phrase that embeds Q but not R. The higher X is in the embedding hierarchy

determiner > pp > gerund > infinitive > finite clause

the stronger the tendency to interpret R as being inside the scope of Q. Conversely, the lower X is in the hierarchy, the stronger the tendency to interpret Q as being inside the scope of R.

The most theoretically interesting aspect of this statement is its symmetry. That is, the rule can not distinguish the case where the embedded np is distributive (the universal quantifier) from the case where the embedded np is nonspecific (the existential quantifier). This symmetry turns out to be tremendously important in the theoretical discussions that follow, and so deserves a closer examination. It turns out that there are several places where the data is not in fact symmetric.

The first asymmetry is apparent in the relative clause data just presented. At the full relative clause end of the embedding hierarchy, one generally finds 100% same/per readings with embedded ***each***. With embedded ***a(n)***, one would expect 100%

different/per readings. But in fact, the preference is consistently less -- hovering near two thirds. One natural text corpus had six full relative clauses with plural heads and embedded indefinite nps. Of these six, only four had a different/per reading. Thus, for nonspecifics, the embedding hierarchy seems to run from 0% to 66% different/per, while for each, it runs from 0% to 100% same/per.

Another asymmetry occurs when the embedding construction does not modify the distributive np. Nominalizations and complements are embedding structures of this kind. In these constructions, the form of the embedding constituent has little effect on the preference for per/relations. If it has any, it is the opposite of that predicted by the hierarchy! In the following example, the form of the verb phrase complement is varied.

(48a) 88% Each secretary reminded me about the scheduling of an appointment.
(48b) 45% Each secretary reminded me about scheduling an appointment.
(48c) 55% Each secretary reminded me to schedule an appointment.
(48d) 16% Each secretary reminded me that I should schedule an appointment.

The embedding hierarchy predicts an even variation: (a) should be close to 0% and (d) should be close to 66%. But the correlation, if any, goes the other way. When *each* is embedded in non-modifying constructions, as in figure 5, the embedding hierarchy correctly predicts the readings. So here there is a clear asymmetry.

Returning to the modifying constructions, one finds a third asymmetry, this time involving the articles of the head np. When *each* is embedded, the article makes little difference in the readings:

(49a) 100% Yesterday at the conference, I managed to talk to a representative from each raw rubber producer in Brazil.

(49b) 100% Yesterday at the conference, I managed to talk to the representative from each raw rubber producer in Brazil.

(50a) 0% Yesterday at the conference, I managed to talk to a guy who is representing each raw rubber producer in Brazil.

(50b) 0% Yesterday at the conference, I managed to talk to the guy who is representing each raw rubber producer in Brazil.

Replacing the nonspecific article *a* with the specific article *the* makes no difference.

This is a consistent counterexample to the generalization that the subject of the different/per relation must be nonspecific.

However, when it is a nonspecific np that is embedded, the articles obey the generalization.

(51a) 0% Striking airline workers forced several major airlines to cancel every flight to an eastern airport today.

(51b) 0% Striking airline workers forced several major airlines to cancel some flights to an eastern airport today.

(52a) 66% Striking airline workers forced several major airlines to cancel every flight which was going to an eastern airport today.

(52b) 0% Striking airline workers forced several major airlines to cancel some flights that were going to an eastern airport today.

Replacing the distributive *every* with the collective *some* destroys the different/per intuition of (52), just as the generalization predicts. So here we have a third asymmetry.

These asymmetries suggest that (47) is not a good way to describe the influence of embedding on quantifier scope. It appears that separate rules will be needed for embedded *each* and embedded *a(n)*.

2.5 Correlations with Surface Order

When neither np is more deeply embedded that the other, as for example when the two nps are clausemates, their surface order seems to be a strong overall correlate of the per intuitions. However, there are many interesting subregularities, as well as a competing analysis that is just as empirically adequate as surface order.

If a distributive np precedes a nonspecific np in the word order of the sentence, then the pair tends to receive a different/per interpretation. If their surface order is reversed, then the pair tends to receive a same/per interpretation. As an example, consider

(53a) 100% The carving of each design from a block of wood
is a requirement of the course.

(53b) 80% The carving of a block of wood into each of these ten designs
is a requirement of the course.

Since ***each design*** and ***a block*** both modify ***the carving***, they are at the same level of embedding. Hence, the embedding hierarchy doesn't apply to this pair. Lexical content alone would lead one to give (b) the different/per interpretation that (a) has. However, the nonspecific np precedes the distributive np. Since this ordering tends to be associated with same/per readings, fewer informants report a different/per reading.

The clausemate nps are the most common example of nps at the same level of embedding. But with clausemates, it is not so clear that surface order is the best correlate of quantifier scope. There is an equally good correlation with the following hierarchy of clausemate positions, which I call the c-command hierarchy:[1]

(55) preposed pp and topicalized np >
subject >
sentential pp and adverbial np >
verb phrase pp >
object

Preposed pps and topicalized nps occur before the subject, and are usually followed by a comma: "For each positive integer, a unique factorization exists." A sentential pp modifies the whole sentence, while a verb phrase pp modifies only the verb phrase -- a distinction which is often too subtle to disambiguate. In general, the verb phrase pps precede the sentential pps in a clause. Hence, this hierarchy differs from surface order only in the last three places. That is, a hierarchy based on surface

---

1. Tanya Reinhart defined the idea of c-command and used it to reformulate Lasnik's Non-coreference rule (see section 4). She also claimed that c-command is better than surface order in predicting quantifier scope judgments. However, she was forced to calculate c-command with respect to the pp containing the quantified np, when there is such an pp, rather than the np itself (see Reinhart 1976, footnote 11, page 209). This modification results in a three layer hierarchy:

(56) preposed sentential pp, left dislocated np >
topicalized np, preposed verb phrase pp, subject, sentential pp >
object, verb phrase pp

The c-command hierarchy given above is a refinement of Reinhart's hierarchy.

order would be

(57) preposed pp and topicalized np >
subject >
object >
verb phrase pp >
sentential pp and adverbial np

The order of the last three items is the reverse of their order in the c-command hierarchy.

The interpretation of the c-command hierarchy is as follows: If a distributive np is higher on the hierarchy than a nonspecific np, a different/per reading is predicted. If the nonspecific np outranks the distributive np, the same/per reading results. If both nps have the same rank in the hierarchy, then a different/per relation is predicted. Note that surface order doesn't matter in this case -- eg. reversing the order of two verb phrase pps will not affect the quantifier scope judgments.

The statistical data support both c-command and surface order equally well. The top part of figure 6 shows the correlation of per judgments with surface order. Of 50 clausemate np pairs, surface order correctly predicted the readings of 42. The bottom part of the figure shows the correlation of the same judgments with the c-command hierarchy. C-command also correctly predicts 42 out of 50 judgments (but not the same 42, of course). Thus, the statistical data that I have gathered doesn't decide the issue.

Unfortunately, paraphrastic data is similarly indecisive. For example, the judgments on the familiar examples

(58a) 50% Ron talked to each woman about a problem.
(58b) 50% Ron talked about a problem to each woman.

(59a) 75% Ron talked to a woman about each problem.
(59b) 80% Ron talked about each problem to a woman.

are independent of surface order. Since both pps are verb phrase pps, the two nps are on the same rank of the hierarchy. Hence c-command successfully predicts the

Fig. 6. Correlations with Clausemates

Across: the position of the nonspecific np (*a(n)* and cardinal articles)
Down: the position of the distributive np (*each* and *every*)
N:M means N different/per readings and M same/per readings

A perfect correlation would be indicated by all entries above the diagonal being N:0, and all entries below the diagonal being 0:M.

Surface Order:

	preposed pp & np	subject	object	verb phrase pp	sentential pp & np
preposed pp & np	-	5:0	4:1	1:0	-
subject	-	-	14:1	11:1	1:1
object	-	0:1	-	1:1	-
verb phrase pp	-	0:2	1:0	-	-
Sentential pp & np	-	1:0	1:0	-	2:0

C-command:

	preposed pp & np	subject	sentential pp & np	verb phrase pp	object
preposed pp & np	-	5:0	-	1:0	4:1
subject	-	-	1:1	11:1	14:1
sentential pp & np	-	1:0	2:0	-	1:0
verb phrase pp & np	-	0:2	-	-	1:0
object	-	0:1	-	1:1	-

data. On the other examples, such as

(60a) 66% The club president splashed a glass of champagne over each member.
(60b) 82% A glass of champagne was splashed over each member by the club president.

the c-command hierarchy predicts a dramatic decrease in the strength of the different/per reading as the nonspecific np is moved from object to subject, that is, from beneath the distributive np's rank to above it. But this dramatic decrease is not evident in the data. On the other hand, the relative surface order of the nps has not changed. Thus surface order correctly predicts the similarity in judgments. In short, paraphrastic data doesn't decide the issue either.

Surface order has a practical advantage over c-command. C-command depends critically on the details of constituent structure, whereas surface order does not. In particular, it is difficult to know whether to attach a pp to the clause or to the verb phrase. Hence, it is not always clear which prediction the c-command hierarchy is making, because the syntactic analysis is ambiguous. Surface order, however, always makes unambiguous predictions.

The symmetry issue is complicated with clausemates because it is the <u>relative</u> order (or rank) that matters. Thus, asymmetric rules, such as

(61) If the distributive np precedes the nonspecific np, then they have a different/per reading; otherwise, they have a same/per reading.

makes the same predictions as the following symmetric rule, written in terms of predicate calculus:

(62) The order of nesting of quantifiers in the logical form is the same as the relative order of the corresponding nps in surface structure.

This rule is symmetric since it doesn't distinguish the existential from the universal quantifier, and it doesn't distinguish the different/per from the same/per reading. But both rules make the same prediction for the general correlation of quantifier scope and clausemate np positions. Because the correlation is founded on the relative order (or rank) of the nps, it can't help decide the symmetry issue.

2.6 *Summary*

This section has reviewed three basic influences on quantifier scope judgments. First, the contribution of articles can be described in terms of the collective/distributive distinction, and the specific/nonspecific distinction. These two distinctions are related to quantifier scope judgments by the following rule:

(63) If a pair of nps receives the different/per reading, then the object of the relation must be distributively interpreted, and the subject must be nonspecifically interpreted.

The distributive hierarchy,

(64) each > every > all of the > all the > other plural articles

neatly correlates the articles of an np with its tendency to take the distributive interpretation, rather than the collective interpretation. No such hierarchy exists for the specific/nonspecific distinction. Instead, ad hoc rules are necessary -- eg. ***a(n)*** is usually nonspecific, definite articles are almost always specific.

The second major correlation concerns structures that embed an np. When one np is more deeply embedded than another, the category of the embedding node determines how easily the embedded np's quantifier may include the other np's quantifier in its scope. But this tendency is asymmetric -- it seems to matter whether the embedded np is distributive (ie. has a universal quantifier) or nonspecific (ie. existential quantifier).

For embedded distributives, the categories fall into a neat hierarchy,

(65) determiner > pp > gerund > infinitive > finite clause

which is called the embedding hierarchy. The higher the embedding structure lies on the hierarchy, the greater the tendency for the embedded quantifier to scope the non-embedded quantifier. Conversely, the lower a form on the hierarchy, the greater the tendency for the non-embedded quantifier to contain the embedded quantifier in its scope.

For embedded nonspecifics, the embedding hierarchy describes the correlation when the embedding construction modifies the distributive np. But when the embedding construction does not modify the distributive np, there is very little correlation of quantifier scope with the form of the construction.

Third, when neither np is more deeply embedded than the other, the order of the nesting of their quantifiers is the same as their relative surface order. However, the quantifier order is equally well correlated with relative rank on the c-command hierarchy:

(66) preposed pp, topicalized np >
subject >
sentential pp, adverbial np >
verb phrase pp >
object

Because it is the relative surface order (or c-command rank), it is impossible to judge whether this correlation is symmetric or asymmetric.

3. *A Transformational Theory of Quantifier Scope*

This section examines the proposal that certain rules governing syntactic transformations also predict the quantifier scope correlations. The following two sections examine the similarity of quantifier scope to pronominal coreference and to lexical composition.

The importance of this exercise is not to find the most empirically adequate description of quantifier scope. As will be seen shortly, the description given in the previous section is much more accurate than any of the theories to be presented. The point is to find the theory with the most independent evidence for its rules. In this way, one comes closer to describing deeper processes, processes that cause both syntactic structures and quantifier scope judgments to have the form they do.

3.1 The Conditions on Proper Binding and Subjacency

Robert May has formulated a theory of quantifier movement within the framework of Chomsky's trace theory (May 1977). Trace theory differs from older versions of transformational grammar in that the transformations are extremely simple, but are subject to constraints that prevent generation of ungrammatical surface structures. For example, the rule that forms WH questions is stated as

(67) Move WH into COMP

WH matches phrases like ***which idiot***, ***whose uncle***, ***in which hand***, etc. COMP is short for "complementizer", a node, usually empty, that immediately precedes the subject of every clause.

To prevent such derivations as

(68) [WH idiot] told Chicken Little the sky was falling.
-->
[] told Chicken Little [WH idiot] the sky was falling.

where the WH np is moved into a COMP node that is lower than itself, one invokes the Condition on Proper Binding:

(69) Condition on Proper Binding
Every variable filling an argument position of a predicate must be properly bound.

(70) Properly Bound
A variable is properly bound by a binding phrase X if and only if it is c-commanded by X.

(71) C-command
A phrase X c-commands a phrase Y if and only if every branching node (ie. a node with more than one daughter) that dominates X also dominates Y, and X does not dominate Y.

Although we will be more interested in how the Condition on Proper Binding effects quantifier scope, it will be illustrated with the WH example above.

Move-WH is a rule that maps deep structure into surface structure. The "variable" of the Condition on Proper Binding refers to the "trace" which, it is postulated, is left behind whenever movement rules operate. Traces are "bound" to the moved phrase. As an illustration of these definitions, consider the two possible surface structures resulting from the application of move-WH on deep structure (a) of figure 7. In (b), the WH np has moved into the main clause's COMP while in (c), it has been moved into the subordinate clause's COMP. In (b), the WH np c-commands its trace. In (c), it does not. Hence, the trace in (b) is properly bound, while it is not properly bound in surface structure (c). The Condition on Proper Binding marks (c) as unacceptable.

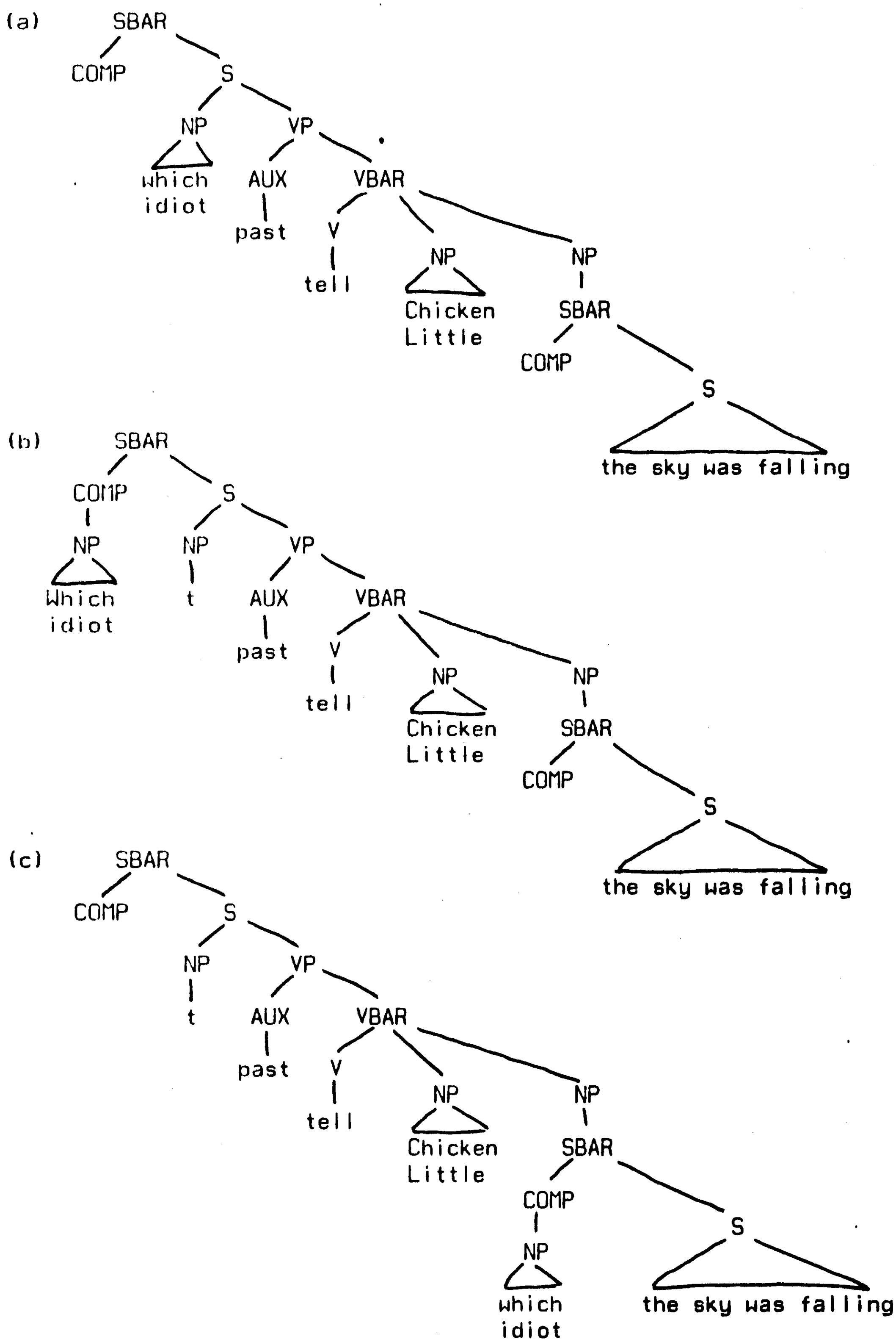
May's rule is simply stated. It is called QR:

(72) Adjoin Q to S.

"Adjoin" means Chomsky adjunction: make a new S node, with Q and S as its daughters. Q matches quantified nps, such as ***some idiot***, ***each egg***, ***two chickens***, ***an exam***, etc. Note that the rule mentions S, the clause's category, explicitly. As will be seen shortly, May's whole theory turns on distinguishing the clause from all other constituent types.

Unlike move-WH, QR maps surface structure into logical form. Essentially, it builds quantifier prefixes for the clauses in the sentence. Figure 8 shows its application. Logical forms (b) and (c) are the results of two possible applications of QR to surface structure (a). May postulates that the Condition on Proper Binding applies to

Fig. 7. The Condition on Proper Binding Constrains WH Movement



logical form as well as surface structure. Hence, (c) is marked as ill formed, since ***some idiot*** doesn't c-command its trace.

In addition to ruling out obviously absurd logical forms like (c), the Condition on Proper Binding accounts for the observations involving quantified nps embedded in a PP. This will be illustrated with the sentence

(73) A representative from each producer spoke with me.

whose surface structure is shown in (a) of figure 9. Since the sentence has two quantified nps, ***a representative*** and ***each producer***, QR applies twice. But there are no constraints on which np is moved first. Thus, both logical forms (b) and (c) can be generated. Here, the bindings of the traces are indicated by coindexing. NP_1 c-commands t_1 in both (b) and (c), but NP_2 c-commands t_2 only in (b). Hence, the Condition on Proper Binding rules out (c) as a possible interpretation of (a).

One might ask what expressions (b) and (c) mean. It turns out that if one ignores the syntactic categories, and concentrates only on the branching structure, May's logical form is a form of typed predicate calculus. In fact, it is nearly identical to the one used by Woods in the LUNAR system (See the appendix, and Woods 1977). In expression (b), NP_2 c-commands NP_1, so ***each producer*** includes ***a representative*** in its scope. Hence, it is predicted that (73) will receive an unambiguous interpretation, with a different representative per producer, which is in fact the case. Thus, a constraint on syntactic transformations accounts for a semantic correlation, namely the PP extreme of the embedding hierarchy.

The other extreme of the embedding hierarchy involves subordinate clauses instead of subordinate pps. To account for this data, May invokes the Subjacency Condition. This constraint upon syntactic transformations can be stated graphically as:

Fig. 8. Subordinate Clauses and QR

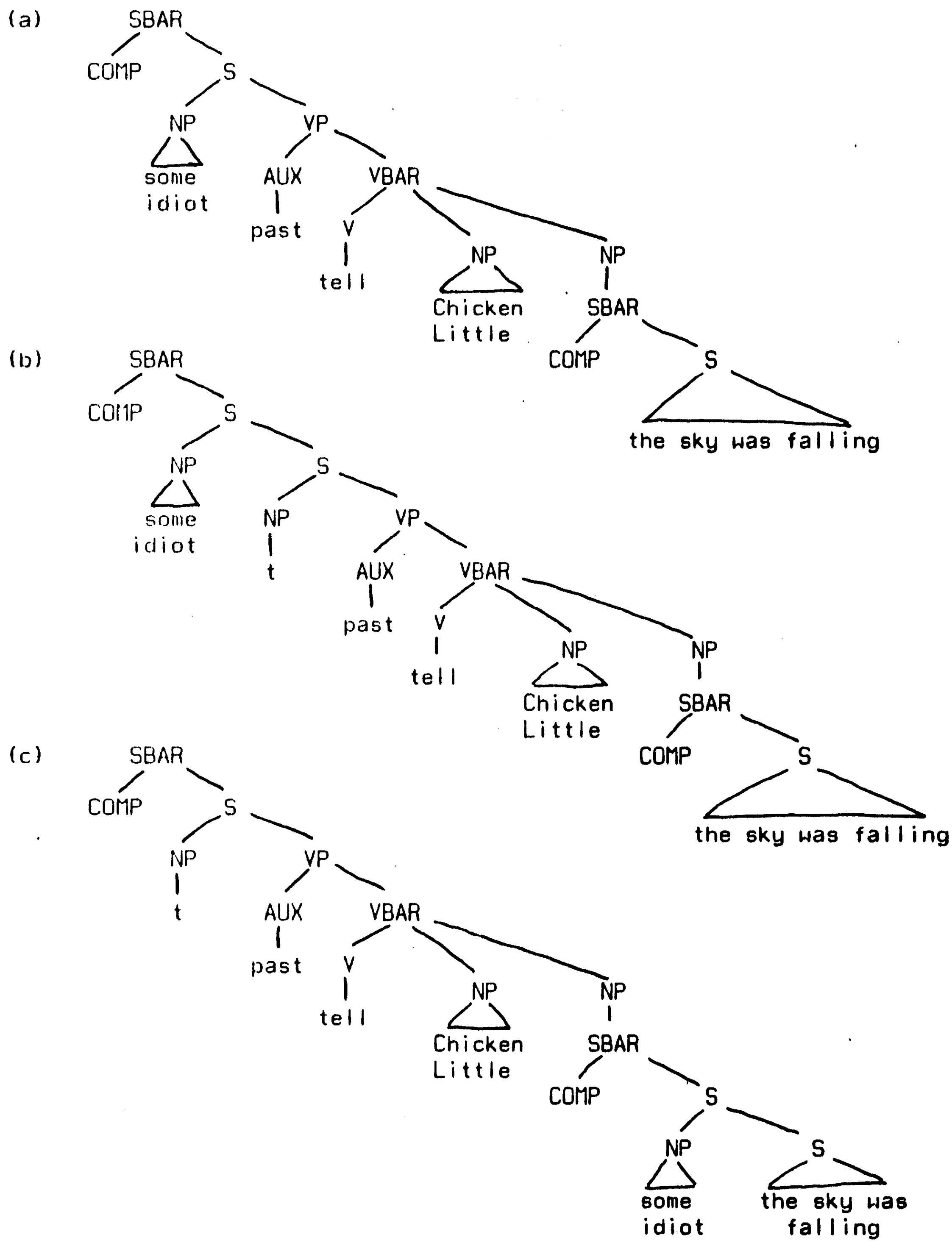
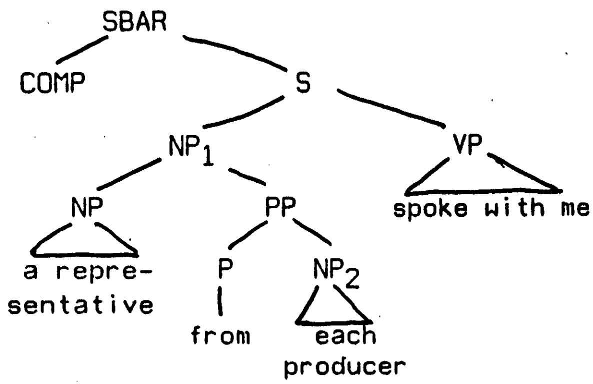
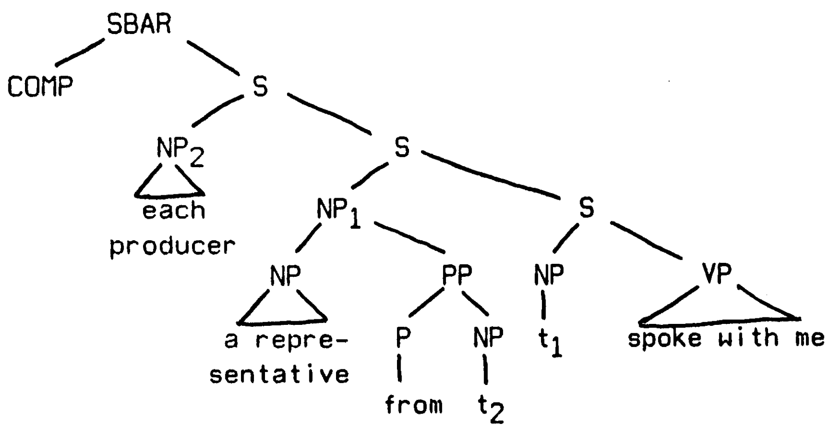


Fig. 9. The Condition on Proper Binding Constrains QR

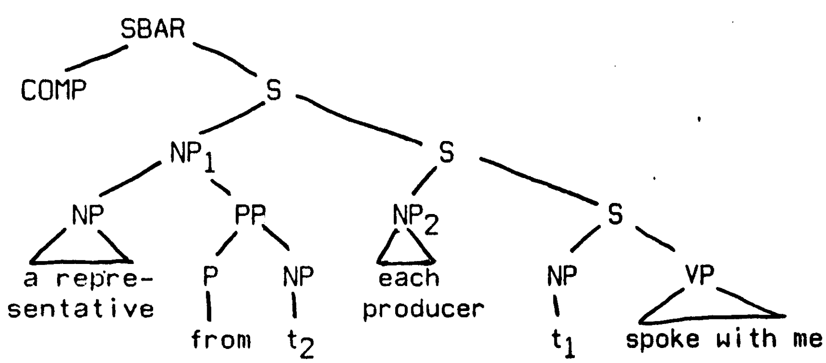
(a)



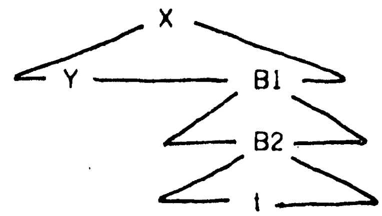
(b)



(c)



(74) The Subjacency Condition
The following structure is unacceptable:



where B1 and B2 are bounding nodes, Y binds the trace t, and there is no trace bound to Y in B1.

For syntactic transformations, a bounding node is usually taken to be any S or NP node. May postulates that, in the translation from surface structure to logical form, S is the only bounding node. If NP were a bounding node, then the expression (b) of figure 9 would be ruled out by the Subjacency Condition. This is a crucial point, one that will be returned to in a moment: May's account of the embedding hierarchy rests on the distinction between the S and NP categories.

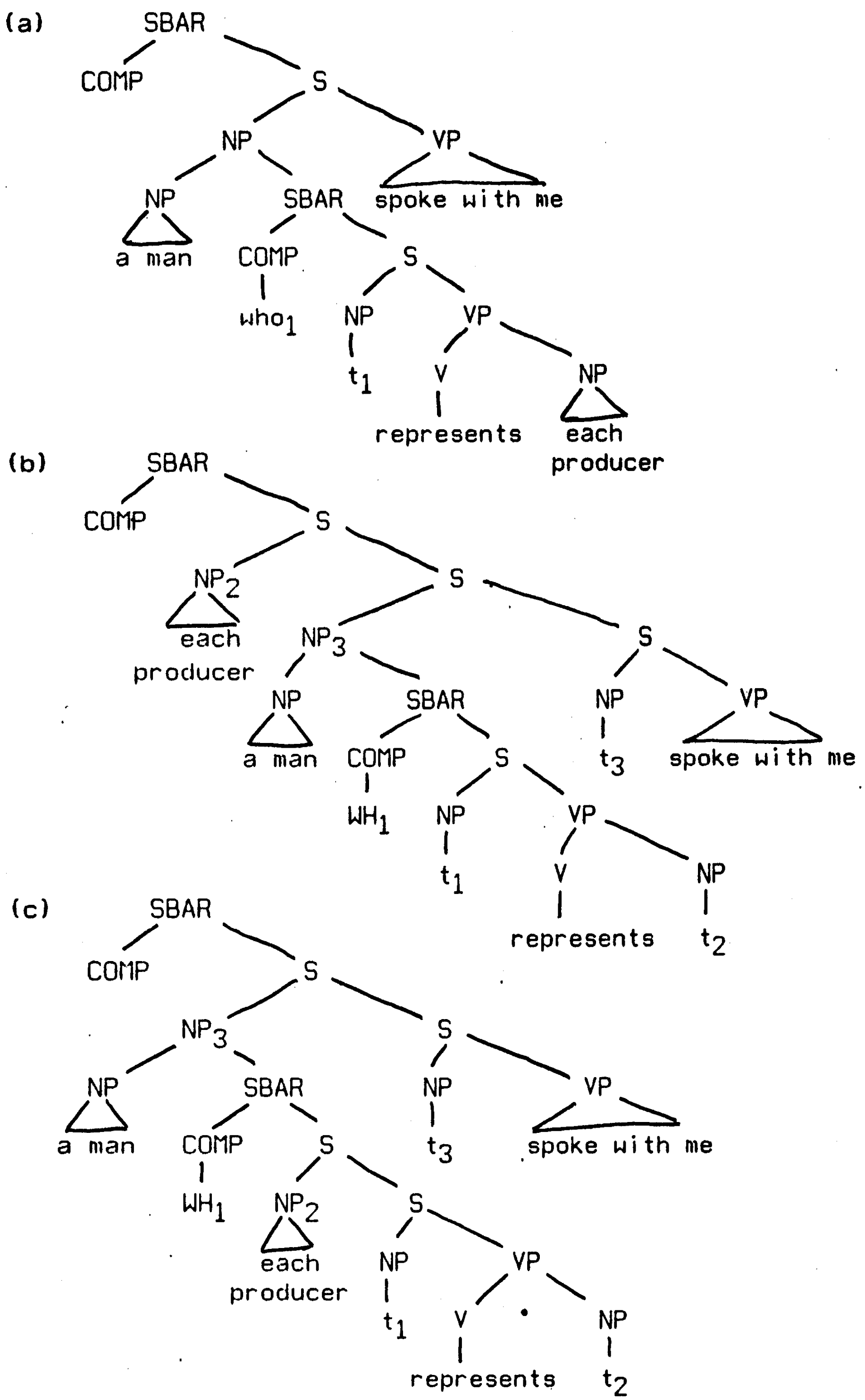
In May's theory, subjacency is responsible for the unambiguous reading of

(75) A man who is representing each producer spoke with me.

The surface structure of this sentence is illustrated in (a) of figure 10. The two logical forms that QR can produce are (b) and (c). The difference between (b) and (c) lies in the location of NP_2. In (b) it c-commands NP_3, which would result in a different/per reading, while in (c) it is adjoined to the subordinate S. In (b), there are two S nodes between NP_2 and t_2 while in (c), there is just one. Hence, Subjacency will rule (b) out, but not (c). This predicts that (a) is unambiguous, with the interpretation that the same man represents all the producers, which is in fact the correct prediction. In short, Subjacency explains why "quantification is generally clausebound", as the old slogan has it (Chomsky 1976).

With two constraints from syntax, May correctly predicts the extremes of the embedding hierarchy. What can be said about the middle, eg. reduced relative clauses and gerunds? A completely adequate treatment would predict that when the embedding constituent has the shape of a verb phrase, then the judgment is ambiguous. Unfortunately, May's approach predicts an unambiguous interpretation.

Fig. 10. Subjacency Constrains QR



Which of the two interpretations -- like a subordinate PP or like a subordinate clause -- depends on whether a reduced relative clause is analyzed as a bare verb phrase, or as a clause with a null subject.

Consider the reduced relative clause

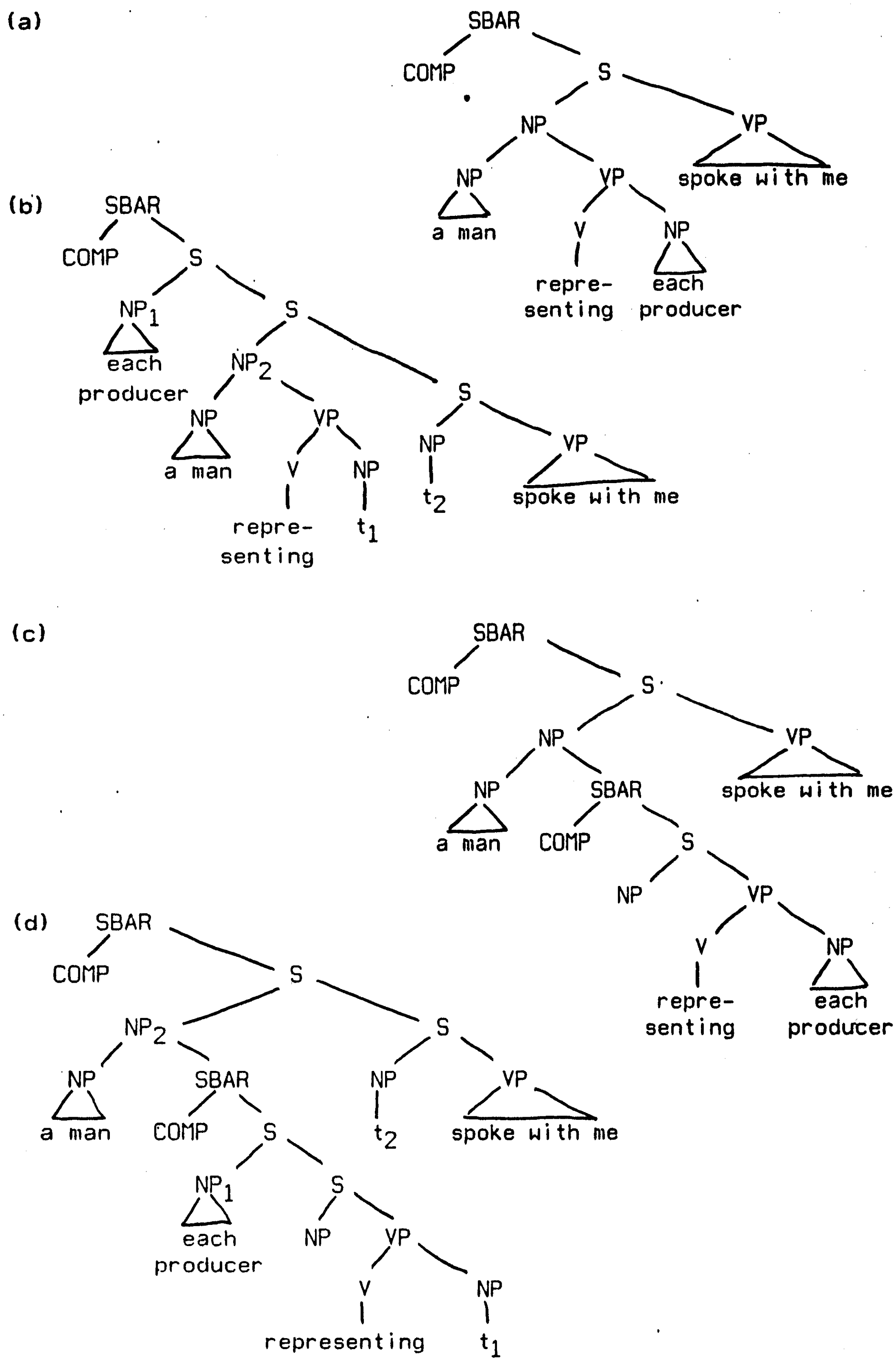
(76) A man representing each producer spoke with me.

If the reduced relative clause is analyzed as a verb phrase, as in (a) of figure 11, then it can only have the logical form shown in (b), since QR states that Q must be adjoined to an S -- adjoining to a VP will not do. Hence, both the embedded np and the np modified by the reduced relative clause are adjoined to the only S there is, namely the matrix S. The Proper Binding Condition forces the two to be nested as shown. Hence, the bare verb phrase analysis of reduced relatives predicts an unambiguous different/per reading. The derivation exactly parallels the translation of the subordinate PP construction.

However, when the reduced relative clause is analyzed as a clause with a null subject, as shown in (c), then Subjacency forces an unambiguous same/per reading, just as it did with full relatives. (d) shows the logical expression that is output.

May's approach, because it relies on the category S both in the statement of QR and in the definition of Subjacency, can only represent ambiguity involving embedded quantifiers by the appearance or non-appearance of an S node. That is, he must introduce a syntactic ambiguity to capture a quantifier scope ambiguity. Thus, whenever an informant reports that, say, a reduced relative clause has an ambiguous quantifier scope interpretation, the reduced relative would have to be given an indeterminant syntactic analysis. This forces syntacticians back to the position held by some descriptive grammarians, that gerund phrases are "half np, half clause", even in a single individual's grammar. Such a consequence is rather unwelcome.

Fig. 11. Two Analyses of a Reduced Relative Clause



3.2 *Asymmetry*

It was noted in section 2.4 that embedding a distributive np is not the same as embedding a nonspecific np. When the embedding constituent is a clause, as in

(77a) 0% Yesterday at the conference, I managed to talk to a guy who is representing each raw rubber producer in Brazil.

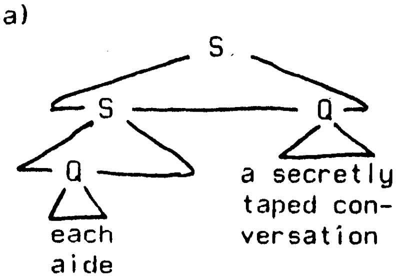
(77b) 0% That each aide knew about the hush money was proved with a secretly taped conversation.

(78a) 66% Striking airline workers forced several major airlines to cancel every flight which was going to an eastern airport today.

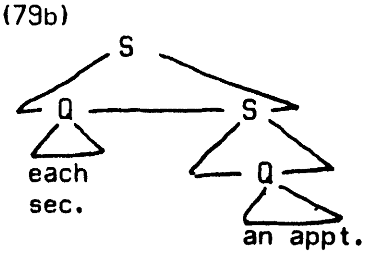
(78b) 88% Each secretary reminded me that I should schedule an appointment.

the informants have unambiguous same/per readings with (77), and ambiguous readings with (78). But May's theory predicts unambiguous readings for both (77) and (78). To see why, consider the following schematic surface structures for the (b) sentences:

(79a)

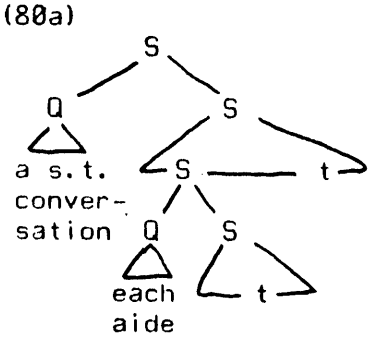


(79b)

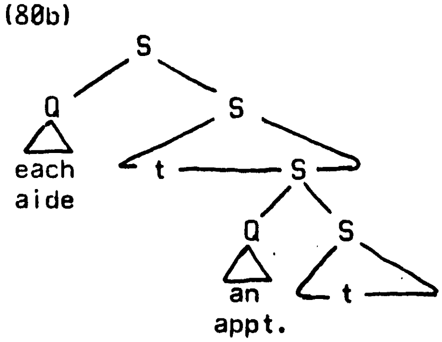


Subjacency prevents the lower quantifier from moving into the upper clause. Hence, the only possible logical forms are:

(80a)



(80b)



Since the upper quantifier c-commands the lower one, the theory predicts unambiguous per/relations. This is correct for (a), but (b) should be ambiguous.

The only way out is via article interpretation rules. To explain away the same/per readings of (b), May could claim either that ***each secretary*** is collective or that ***an appointment*** is specific. But this is a very powerful use of the article interpretation rules. In fact, it is possible to do away with half the quantifier movements, and use article interpretation to take up the slack. That is, the new QR rule would raise only universal quantifiers, like ***each***, and leave all other nps untouched. Whenever the raised ***each*** np c-commands an indefinite, but the sentence lacks a different/per reading, one would claim that the indefinite np has the specific interpretation. A theory that is very similar to this is presented in section 5.

3.3 *The Interaction of QR and WH*

May chose not to model the influence of surface order on quantifier scope. That is, when no np is more deeply embedded than the others, QR is unconstrained and generates all the possible quantifier scope nestings. May claims that surface order predicts the "preference" of one quantifier scoping over another, but that QR predicts the "markedness" of one quantifier scoping over another (see his footnote 14, chapter 1). However, a comparison of the statistics in figures 3 and 4 shows that the correlation of quantifier scope with surface order is somewhat tighter than its correlation with the embedding hierarchy. Whether the distinction between "preference" and "markedness" can stand in the face of such facts remains to be seen.

There is, however, one case where May's approach does make a prediction: a WH np is predicted to be outside of the scope of any of its clausemates. Consider the sentence

(81) Which city has each burglar been assigned to?

whose surface structure appears in (a) of figure 12 (ignoring the passive). The only logical form this sentence can have is (b). Because QR adjoins to S, and Move-WH fills the COMP node, which always c-commands S, WH nps are predicted to be unambiguously outside the scope of all quantified clausemate nps. But this prediction can not be validated.

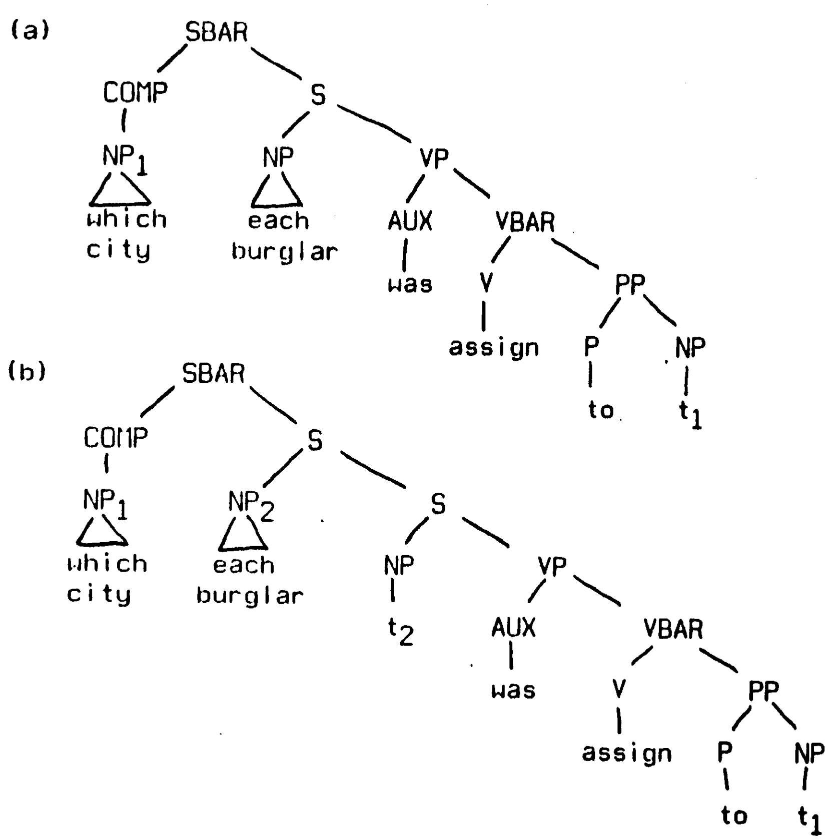
In testing the sentences

(82a) Which city has each of the burglars been assigned to?
(82b) Which state has each presidential candidate spent the most money on?

a dialect split was evident. Half the informants felt the sentences were just fine, with the different/per reading, where *each* containes ***which*** in its scope. That is, they understood the sentence as asking for a list. This is the opposite interpretation from the one predicted by May's theory. The other informants rejected the sentences. All of them complained that it was clear that the sentence was asking for a list of cities or states, but they objected to the phrasing of the request. This indicates that they had a pragmatic preference for the different/per reading, but a linguistic process was blocking this reading. The interpretation of these informants

---

Fig. 12. Interaction of Move WH and QR



supports May's theory.

But a simple dialect split is not the end of the story. When the WH questions were embedded,

(83a) Woodward wanted Bernstein to find out which city each of the burglars had been assigned to.

(83b) Woodward wanted Bernstein to find out which state each presidential candidate had spent the most money on.

the dialect split disappeared! All of the informants found the sentences quite acceptable, and gave them a different/per interpretation. This result is opposed to the predictions of May's theory. Moreover, counterexamples to May's claim even occurred in natural text:

(84a) This knowledge breaks down into subcategories according to just what time specifications are present on each instantiated frame.
-- different time specifications per frame

(84b) The following schematic definitions for descriptions show what properties they can have, and what kinds of values each property can take.
-- different value types per property

In both sentences, the WH is inside the scope of the *each*[1]. Hence, May's claim can be refuted with naturally occuring counterexamples. Section 5 proposes a theory that accounts for this dialect split, and its disappearance when the WH clause is embedded.

May supports his analysis of the WH/QR interaction by noting that sentences like (a) below sound much worse than (b)

(85a) * Which men in some city voted for Debs?
(85b) Which men in Cleveland voted for Debs?

He notes that (a) can not receive an interpretation by his rules -- the only way to adjoin ***some city*** to S leaves it below ***which men in t***, and hence the Proper Binding Condition will mark the interpretation as unacceptable. However, all of May's examples involve an indefinite quantifier. When definite quantifiers are embedded

1. Although these WH phrases are the heads of free relative clauses, not embedded WH questions, they are still dominated by COMP, in the current version of trace theory.

under the WH, as in

(86) Which men in each city voted for Debs?

the sentence is fine. This leads one to speculate that some functional explanation, such as those promulgated by Kuno and the Prague linguists (Kuno 1975) might account for the unacceptability of (85a).

3.4 Summary

May's theory of quantifier scope is an important step forward. It uses two well motivated rules of trace theory to account for the embedding hierarchy, a phenomena that has previously been captured only with unmotivated, a posteriori rules. The movement from description to theory, or if one prefers, to simpler, more encompassing descriptions, is always welcome, since it paves the way to causal explanations.

The fact that May's theory ignores the influence of surface order on quantifier scope should probably not be held against it. Since it allows all possible readings, a surface order rule could be added to the theory to rule out the non-occuring readings. However, it would probably be difficult to motivate such a rule, since transformational grammarians have traditionally been reluctant to incorporate surface order into their rules.

May's theory has a grave defect. Since the theory turns on distinguishing S from the other categories, it is difficult to capture the ambiguity of the middle of the embedding hierarchy. That ambiguity could only be captured by introducing a syntactic ambiguity. There is second problem with reliance on S -- it predicts that a WH np is always outside the scope of any of its clausemates' quantifiers. But this prediction is not veracious.

Lastly, the theory predicts that embedding an *each* np should be symmetric with embedding an *a(n)* np. Since this is not the case, a powerful article interpretation rule would have to be added to create the necessary asymmetry.

4. *An Anaphoric Theory of Quantifier Scope*

The idea that quantifier scope is a highly abbreviated form of anaphora has a strong intuitive appeal. In the sentences below, one feels similar different/per relations as the anaphora changes from explicit pronominal coreference, through more abbreviated anaphoric constructions and finally arrives at a quantifier scope sentence:

(87a) Each red node is attached to a node to the left of it.
(87b) Each red node is attached to a previous node.
(87c) Each red node is attached to an appropriate node.
(87d) Each red node is attached to a light blue node.

As the anaphora becomes more abbreviated, the pragmatic relation between the two nps becomes less explicit and the reader becomes less certain whether they have a different/per reading. It is clearer in (87a) than in (87b) that each red node is attached to a different node.

If quantifier scope is an abbreviated form of anaphora, one might expect rules that constrain the coreference relation to constrain the per relations as well. Two linguists, Keenan and Reinhart, have argued just exactly that (Keenan 1974, Reinhart 1976). The following account is an amalgamation of their theories. It differs from theirs in that it does not use traditional predicate calculus as the logical form. Instead, it is based on Skolem form, a logical notation that is little known outside of the theorem proving community.

4.1 *Typed Skolem Form*

When Frege invented predicate calculus (Frege 1878), he incorporated into it two basic ideas: First, the function/argument notation of mathematics should be used instead of the subject/predicate notation of Aristotelean logic. Secondly, tree structure and variable-binding operators should be used to explicate the scopes of negations and generalities. Skolem form retains the first idea, but modifies the second. Indeed, it uses the the function/argument notation to replace part of the scope notation.

To convert predicate calculus to Skolem form, one replaces each existentially bound variable by an anonymous function, whose arguments are the variables bound by universal quantifiers that included the existential quantifier in their scope. Thus, (a) in predicate calculus becomes (b) in Skolem form:

(88a) $\exists v\ \forall w\ \exists x\ \forall y\ \exists z\ \ P(v\ w\ x\ y\ z\ )$

(88b) $P(\ f()\ w\ g(x)\ y\ h(w\ y)\ )$

The existential variables v, x and z have been converted to Skolem functions f, g and h.

The basic idea of Skolem form is to link each existential quantifier explicitly to the universal quantifiers that scope it. That is, when the quantifier order is $\forall x \exists y$, y is linked to x. But when the order is $\exists y \forall x$, y is not linked to x.

The linkage is represented with the function/argument relation. That is, when y is linked to x, it is represented as a function with x as its argument. Of course, Skolem functions are in some sense just dummies. Unlike ordinary functions, such as "mother-of" or "square-root", one can't compute the value of a Skolem function from its arguments. Skolem functions are just a mechanism for representing quantifier scope.[1]

Luckily, the anaphoric theory of quantifier scope can be presented without introducing a complex new formal language. Just as May used, as his logical form, a modified surface structure that can easily be converted to typed predicate calculus, this section will use a modified surface structure that can be easily converted to

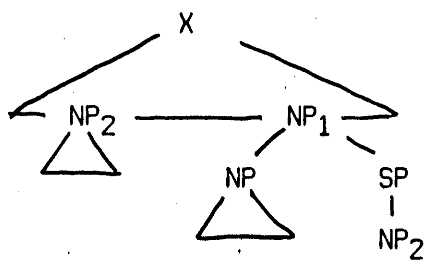
---

1. Most semantic net formalisms have used explicit links to represent quantifier scope, and in that sense can be considered Skolem forms. On this, see Woods 1975, section F.

typed Skolem form. [2]

In particular, the explicit linkage of Skolem form will be indicated by attaching extra nominal modifiers containing dummy nps. If NP1 would be represented as a Skolem function with NP2's bound variable as one of its arguments, then the logical form will be

(90)



The nominal modifier SP (for "Skolem Phrase") dominates an empty np (or "trace" if one prefers) that is coindexed with NP2. The SP node is included to make the structure similar to the possessive and pp modifiers -- a property that will be useful later.

Figure 13 illustrates how this logical form represents the per relations. (b) is the logical form for (a) when it has the interpretation "Each frat brother dated a different woman", since ***a woman*** has a Skolem modifier which is coindexed with ***each frat brother***. Expression (c), on the other hand, lacks the extra modifier. Hence, expression (c) means "All the frat brothers dated the same woman." A more

---

2. "Typed" Skolem form will be used instead of the usual, untyped Skolem form for the same reason that May used typed predicate calculus instead of ordinary predicate calculus: it makes the translation into logical form simpler. The descriptive content of the quantified np is translated into the type function, thus establishing the range of quantification. Translation into untyped logical forms requires introducing sentential connectives. Compare the typed predicate calculus of (b) with the untyped predicate calculus of (c):

(91a)	Each boy kissed a girl.
(91b)	$\forall$x:boy() [$\exists$y:girl() [x kissed y]]
(91c)	$\forall$x [boy(x) $\supset$ $\exists$y [girl(y) & [x kissed y]]]

Note that "boy" is a type function in (b) but a predicate in (c). This makes the translation into (b) much simpler than translation into (c) (cf. Woods 1977). Similar similicity is realized by using typed Skolem form instead of ordinary Skolem form. However, since this report uses modified surface structure as logical form, the distinction between typed and untyped logical forms is peripheral.

precise definition of the meaning of this logical form is the topic of the next section.

4.2 The Semantics of Typed Skolem Form

One way to precisely communicate the meaning of a logical notation is to give it a formal semantics. A formal semantics is an algorithm that, given an expression in the logical form and a model of the world, calculates whether the expression is true in the model. Such a semantics for typed Skolem Form is presented in the appendix. The main insight to be gained there can be summarized in term of the per relations:

(92) If NP1 has a distributive interpretation, and is an *argument of NP2, then the informant will report a different/per relation between NP1 and NP2. Otherwise, the informant will report a same/per relation.

(93) <u>*argument</u>
NP1 is an *argument of NP2 if and only if
(a) NP1 is an argument of NP2, or
(b) NP1 is coindexed with NP3, and NP3 is an *argument of NP2, or
(c) NP1 is an argument of NP3, and NP3 is an *argument of NP2.

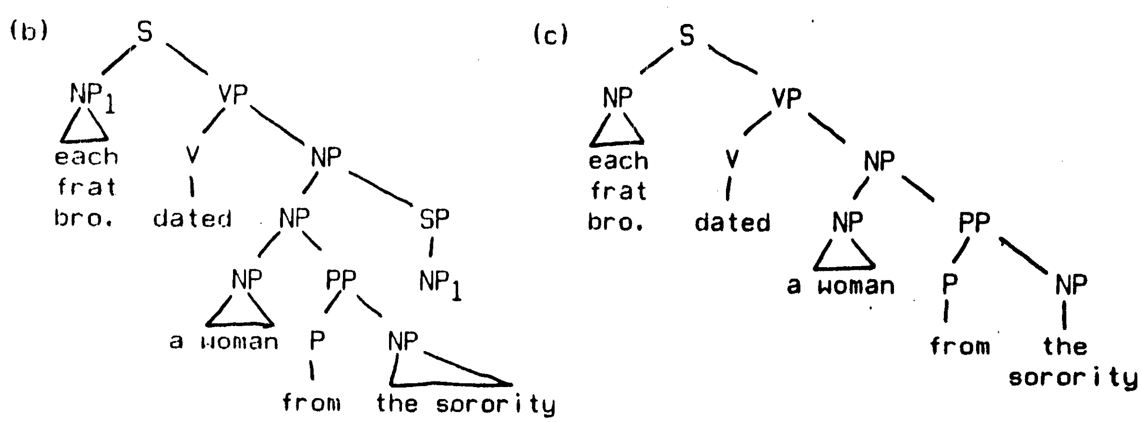
(94) <u>argument</u>
NP1 is an argument of NP2 if and only if it is the object of a pp, possessive or Skolem modifier of NP2.

"*argument" is just the transitive closure of the function/argument relation. Note

---

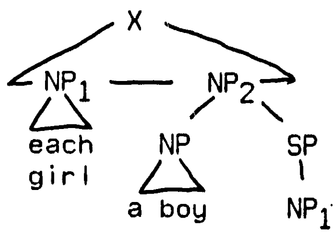
Fig. 13. Typed Skolem Form Expressions for the Per Relations

(a) Each frat brother dated a woman from the sorority.

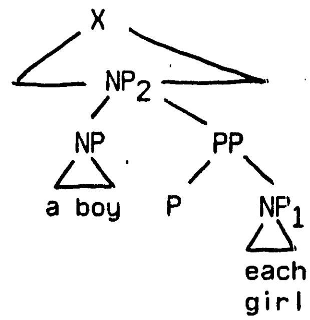


that the semantics of the logical form have been defined so that *argument does not distinguish between empty nps coindexed with a distributive np, and the distributive np itself. Hence, two expressions of the form

(95a)



(95b)



will both have the reading "different boy per girl" since NP1 is a *argument of NP2 in both cases.

The definitions of argument and *argument make an empirical prediction. In fact, they account for part of the embedding hierarchy. Whenever a distributive np is in a pp or possessive modifier of another np, as in

(96) I talked to a representative from each producer.

then the construction has a different/per reading. Although it looks like we have accounted for a correlation "for free", this correlation was in fact a major consideration in designing the formal semantics, which is in turn reflected in the definitions above. Thus, the "meaning" for the typed Skolem form should be subjected to the same empirical scrutiny as a translation rule.

The definition of *argument captures a generalization concerning the article translation rules. It was noted in section 2.1 that specific nps can not be the subjects of different/per relations. This generalization is easily captured with the translation rule:

(97) A specific np can not have Skolem modifiers.

In section 2.4 it was noted that NP-PP constructions are consistent counterexamples to the generalization that specific nps can not be the subjects of different/per relations, since they may have a different/per reading even if the head np is specific.

(98) I talked to $\left\{\begin{array}{l}\text{the}\\ \text{a particular}\\ \text{every}\end{array}\right\}$ representative of each producer.

But when the generalization is accounted for by rule (97), such sentences are no longer counterexamples. Even though ***the representative*** can not have a Skolem modifier, it gets its different/per interpretation via a regular modifier that contains a distributive np.

The definition of *argument makes a strong empirical claim that certain configurations of per relations can never occur. If an NP-PP construction does not have a different/per reading, then the embedded np is not distributive. If it were distributive, then it would necessarily have a different/per reading, since it is a *argument of the other np. This claim can be substantiated with sentences like

(99) A cruise to every Aegean port would require a port pass.

All informants agreed that the same ship was going to all the ports. In addition, they agreed that the same port pass would work for all the ports. Most people pointed out that the latter was rather unusual. They would have expected a different port pass per port, but the sentence simply did not say that.

What seems to be going on here is this. A strong preference for cruises to visit more than one port has forced ***every Aegean port*** to be interpreted collectively. If it were interpreted distributively, then the informants would have a different cruise per port, since the np ***every Aegean port*** is inescapably a *argument of ***a cruise***. Hence, even if ***a port pass*** has a Skolem np modifier that is coindexed with ***every port***, it can not be the object of a different/per relation because ***every port*** is collective. Thus, the assumption that distributive arguments indicate different/per readings accounts for the counter intuitive reading of (99).

Three empirical arguments have been presented that support the definition of *argument and its association with different/per readings. This indicates that the meaning given to the logical form is well motivated.

4.3 Bound Quantifier Scope

The next few sections concentrate on constraints on the translation from surface structure to logical form. It is shown that the rules that constrain anaphora also constrain quantifier scope. In fact, given the logical form introduced above, one need only replace the notion "coreference" with the notion "coindexing" in the rule statements.

From the standpoint of constraints on rules, there are actually two kinds of anaphora in English: bound and unbound. It turns out that there are also bound and unbound versions of quantifier scope.

The paradigmatic cases of bound anaphora are the reflexive pronouns (***herself***, ***himself***, ***itself***, etc.) and the reciprocal construction ***each other***. The following sentences illustrate the constraints on bound anaphora.

(100a)	* Herself slept.
(100b)	* Each other slept.
(101a)	* Mary said that John talks to herself.
(101b)	John said that Mary talks to herself.
(101c)	* The men said that John talks to each other.
(101d)	John said that the men talk to each other.
(102a)	* John talked to herself about Mary.
(102b)	John talked to Mary about herself.
(102c)	* John talked to each other about the men.
(102d)	John talked to the men about each other.

The first two examples, (100), show that a bound anaphoric element (eg. ***herself***, ***each other***) must corefer with something or the sentence is unacceptable. The sentences of (101) show that bound anaphoric elements must be clausemates of their antecedents (i.e. the np that they corefer with). The sentences of (102) show that the antecedent must precede the anaphoric element. These constraints on bound anaphora can be summed up in the following descriptive rule:

(103) If X is a bound anaphoric element, then
(a) X must have an np antecedent, and
(b) the antecedent must be a clausemate of X, and
(c) the antecedent must precede X.

The constraints on bound anaphora are actually much more complex than this (see

Jackendoff 1972), but this rule is a good first order approximation.

The bound form of quantifier scope is marked by placing ***each*** or ***apiece*** after an np. For example,

(104a) The women built two bookcases each.
(104b) The women bought a bookcase apiece.

Suppose that the np that has the ***each*** or ***apiece*** after it, also has a Skolem modifier. The dummy np of the Skolem modifier can be equated with the anaphoric element of bound anaphora. The following examples show that bound quantifier scope, as this phenomena might be called, has the same distribution as bound anaphora.

(105a) Two bookcases each were built.
(105b) * Two bookcases apiece were built.

(106a) * The women said that John built two bookcases each
(106b) John said that the women built two bookcases each.
(106c) * The women said that John built two bookcases apiece.
(106d) John said that the women built two bookcases apiece.

(107a) * John talked about two issues each to the women.
(107b) John talked to the women about two issues each.
(107c) * John talked about two issues apiece to the women.
(107d) John talked to the women about two issues apiece.

The sentences (105) show that the Skolem np must have an antecedent, that is, it must be coindexed with some lexical np. (a) lacks a star because its has a reading where *each* is a quantificational adverb (see Keyser and Postal 1976 on Quantifier Floating). (106) shows that the antecedent must be a clausemate of the Skolem modifier. (107) shows it must precede the Skolem modifier as well. In short, the bound quantification construction obeys rule (103), with "coindexed" substituted for "coreference".

The unacceptability of the starred sentences seems to me to be less pronounced than the unacceptability of the corresponding anaphora sentences. This is consistent with the claim that quantifier scope correlations are epiphenominal. The process that constrains anaphora also constrains quantifier scope, but not as effectively.

4.4 Keenan's Functional Principle and Partial Ordering

The most common form of anaphora is unbound anaphora. The paradigmatic examples are the personal pronouns -- *she*, *he*, *it*, etc. This kind of anaphora is called unbound because the antecedents can be just about anywhere. Indeed, antecedents for some forms of unbound anaphora need never appear explicitly in the text. In the example

(108) Jon wants to meet with you tomorrow. It has to be sometime in the morning, because he's going sailing in the afternoon.

The antecedent of *it* doesn't actually appear in the text.

The current view of unbound anaphora is that there are no rules which force two nps to corefer, but there are rules which block coreference in certain situations. Currently, there are three major rules known to block unbound anaphora. One of them, Keenan's Functional Principle, will be covered in this section. The other two will be discussed in the following section. Together, these three rules are sufficient to account for most of the embedding hierarchy and the c-command hierarchy.

Keenan's Functional Principle is designed to rule out coreference between a function and its arguments. It explains the blocking in

(109a) Some chairs stacked on themselves fell over.
(109b) Some chairs stacked near the room they were removed from fell over.

(110a) * Some stacked chairs on themselves fell over.
(110b) * Some stacked chairs on them fell over.

Although coreference between *them* and *chairs* is allowed in (109), it is blocked in (110) because the object of *on* is understood as an argument of *stacked chairs*.[1]

Keenan's Functional Principle, as stated in Keenan 1974, is much broader than the

---

1. Sometimes a pronoun in a relative clause can't corefer with the head np, as in

(112) * The man$_1$ who the woman he$_1$ loved betrayed -- is despondent.

According to Chomsky 1975, this blocking is the result of the Non-definite anaphora rule, which is discussed in the next section.

version that will be used here. His version is

(113a) The reference of the argument expression must be determinable <u>independently</u> of the meaning or reference of the function symbol.

(113b) Functions which apply to the argument however may vary with the choice of argument, and so need not be independent of it.

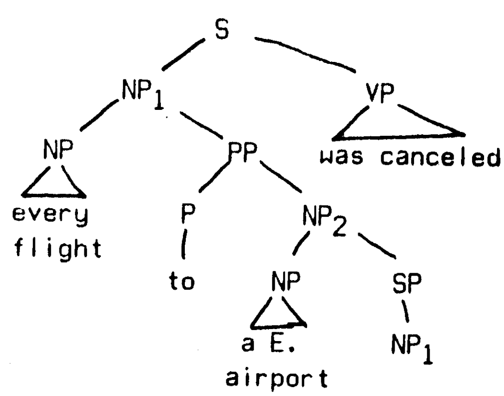
Keenan applies this principle to constructions, such as relative clauses and subject-VP, that are not taken, in this report, to result in function/argument relations in logical form. Thus, the Functional Principle will be taken to be the following very narrow rule:

(114) No np may be coindexed with one of its *arguments.[1]

Besides explaining anaphoric data, the Functional Principle explains part of the embedding hierarchy. That is, when a nonspecific np is embedded in a pp (or possessive) that modifies a distributive np, as in

(116a) Every flight to an eastern airport was canceled.

(116b)



then the Skolem modifiers of the nonspecific np can't be coindexed with the distributive np. That is, if ***an eastern airport*** has a Skolem modifier, it can't be

1. By using "*argument" instead of "argument", the power of the Functional Principle has been extended somewhat. However, this extension stays within the spirit of Keenan's rule. It also explains why the following version of the famous Peters and Richie sentence is unacceptable:

(117) * I talked to [his$_1$ wife]$_2$ about [her$_2$ husband]$_1$.

According to the definition of *argument, both ***his wife*** and ***her husband*** are *arguments of themselves!

coindexed with ***every flight***. As (b) shows, such a Skolem modifier would be a *argument of ***every flight***, so the Function Principle will rule (b) out as a logical form for (a). Hence, the construction can't have a different/per reading, which is in fact the case.

What the Functional Principle actually says is that the function/argument relation is a partial order. No cycles are allowed. Hence, there is no need for a theory of quantification that is more general than a partial order. Jaakko Hintikka (Hintikka 1974) claims that totally ordered theories of quantification, such as predicate calculus, are unable to express the meaning of certain sentences of English. So it seems that a logical form that admits partially ordered quantifier scopes, as Skolem form does, is both necessary (Hintikka) and sufficient (Keenan) for English.

If Hintikka were right, this would be a strong argument for the anaphoric theory over the transformational one. As it happens, there is a flaw in his argument. The rest of this section is a critique of the argument. Since it turns out to be inconclusive, the reader may wish to skip to section 4.5. The argument, and its rebuttal, are interesting examples of empirical arguments that bear directly on the expressive power of logical form.

Hintikka claimed that standard first order logic cannot represent the quantifier scope reading of the following sentence.

(118) Some book by every author is referred to in some essay by every critic.

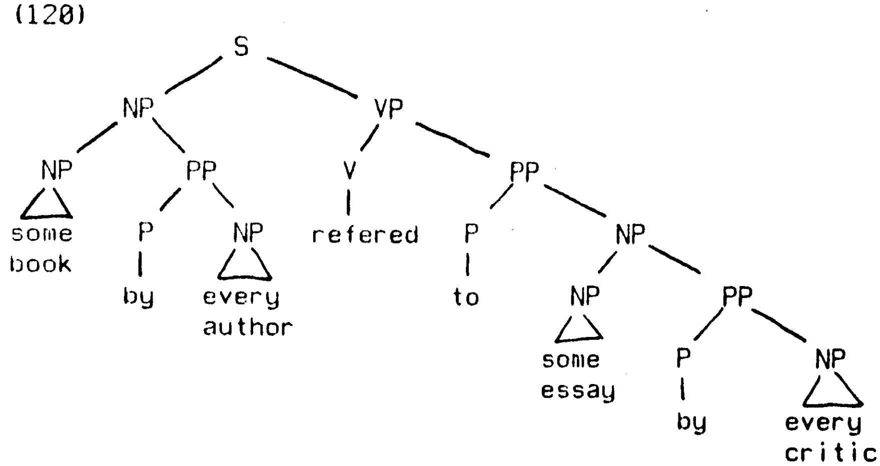
The crucial intuition here is that the choice of the essay is independent of the author, and that the choice of the book is independent of the critic. Thus, ***every author*** should be outside ***some book*** but inside ***some essay***. ***Every critic*** should be outside ***some essay*** but inside ***some book***. It isn't possible to lay out the four quantifiers in a line and preserve these intuitions.

Standard logic indicates that the choice connoted by the $\exists$ operator is independent of an $\forall$ operator by writing the $\exists$ operator outside the $\forall$ operator. Hintikka proposed a second method of indicating the independence of $\exists$: write the $\exists$ above or below the $\forall$. Thus he would write the representative of (118) as

(119) $$\left.\begin{matrix}\forall a\ \exists b \\ \forall c\ \exists e\end{matrix}\right> [\text{author}(a)\ \&\ \text{book-by}(b\ a)\ \&\ \text{critic}(c)\ \&\ \text{essay-by}(e\ c)] \supset \text{refers-to-in}(b\ e)$$

Hintikka calls this logic "finite partially-ordered quantification theory". Linguists often refer to it as "branching quantifiers". But as Hintikka points out, it is equivalent to Skolem form. The typed Skolem form expression that represents this reading is:

(120)



Note that no Skolem modifiers are necessary. The two different/per relations arise from the fact that the two distributive nps, ***every author*** and ***every critic***, modify the two indefinite nps.

There is some dispute over Hintikka's intuition that (118) must have the branched interpretation. Gilles Fauconnier presented his informants with the sentence, and various factual contexts (Fauconnier 1975). He then asked whether the sentence was true in each of the contexts. His informants felt that the choice of the essay could be different with different authors. He reports "Speakers were apparently satisfied that if for any pair (author, critic) a corresponding pair (book, essay) could be found, sentence (118) was true." That is, only when a context violated the weakest possible reading for the sentence -- $\forall\forall\exists\exists$ -- would the context make the sentence false. Similar results were obtained with other sentences of roughly the same form.

Fauconnier's test, I think, determines only whether the sentences MUST have a branched reading. His test uncovers only the weakest reading. Using the usual interview technique, I found two sentences whose most prominent reading is a branched one. They are

(121a) Run a wire from a bit in each memory to an alarm in each room.
[3] branched
[2] (∀ memory) (∃ bit) (∀ room) (∃ alarm)
[1] (∀ memory) (∀ room) (∃ bit) (∃ alarm)
[1] (∀ room) (∃ alarm) (∀ memory) (∃ bit)

(121b) A biography of each Lake poet was referred to in a talk by each Phd candidate.
[5] branched
[1] (∃ biog) (∀ candidate) (∃ talk) (∀ poet)
[1] (∀ candidates) (∃ talk) (∀ poet) (∃ biog)
[1] branched on 1st reading, ∀c∃t∀p∃b on 2nd

Informants were asked how many wires, bits and alarms they should need to accomplish this command, given that there are three computer memories and three rooms. A majority of the informants reported that they would need three bits, three alarms and nine wires. That is, they interpreted the command as requiring one bit per computer, one alarm per room, and enough wire to connect every bit to every alarm. This indicates their prefered interpretation is the one that can't be represented in standard logic.

These data seem to indicate that one would use a representation as powerful as Skolem functions if one wishes to represent the preferred readings of all sentences, but that one might be able to get by with standard logic if only the weakest reading of a sentence is important.

However, if predicate calculus is augmented with a nonstandard operator to represent the specific interpretation, then the branched readings can be represented. That is, the branched readings correspond to the major readings of

(122a) Run a wire from the power-glitch bit in each memory to the system crash alarm in each room.

(122b) The standard biography of each Lake poet was referred to in the thesis defense of each PhD candidate.

where the indefinite articles have been replaced by the specific article ***the***. If S is the new specific quantifier, then (121a)'s most popular interpretation could be represented as

```
(123)     (∀m : memory()
            (Sb : bit(m)
              (∀r : room()
                (Sa : alarm(r)
                  (∃w : wire()
                    (Run w from b to a))))))
```

The formal semantics of the specific indefinite quantifier S are given in the appendix.

The flaw in Hintikka's argument lies in the fact that the ***each*** nps are arguments of the indefinites in all the branched interpretation sentences he cites. By using the specific indefinite quantifier, which is insensitive to universal quantifiers that scope it, one can get around the necessity of partially ordered quantification. For an airtight argument, Hintikka would have to find a clause with four nps, none of which modify the others.

In short, Hintikka's sentences argue either for partially ordered quantification, or for inclusion of the specific indefinite operator in predicate calculus.

4.5 Non-coindexing Rules

One coreference constraint, the Functional Principle, has been shown to constrain quantifier scope. This section discusses the other two constraints on coreference.[1] There are many versions of these two rules in the literature, The most recent versions, due to Tanya Reinhart (Reinhart 1976), are

---

1. There is a third non-coreference rule which will not be discussed. The Disjoint Reference rule, discussed in Chomsky 1976, is the converse of the reflexive rule. That is, if X and Y are clausemates, and X precedes Y, and Y is not a reflexive pronoun, then they can't corefer. For example,

(126) * $John_1$ talked to $John_1$.

That is, unbound coreferpnce is ruled out exactly where bound coreferenc would be permitted. Note that there is no analogous rule for reciprocals or quantifier scope:

(127a) Each of the men_1 talked to the $others_1$.
(127b) Each man talked to a woman.

Hence, the Disjoint Reference rule is unique to coreference.

(127c) The Non-coreference Rule
If X and Y are nps such that
X c-commands Y in surface structure, and
Y is not a pronoun,
then X and Y can not corefer.

(127d) The Non-definite Anaphora Rule
If X and Y are nps such that
X is a non-definite np, and
X does not c-command Y in surface structure,
then X and Y can not corefer.

(127e) C-command (repeated from section 3.1)
A phrase X c-commands a phrase Y if and only if every branching node (i.e. a node with more than one daughter) that dominates X also dominates Y, and X does not dominate Y.

(127f) Non-definite Nps
An np is non-definite if it has the articles ***each, every, all, or no***; if it is non-specific; if it receives contrastive stress; of if it is the trace of WH movement.

The Non-coreference Rule accounts for "backwards pronominalization" paradigms, such as the following:

(129a) * $Nixon_1$ hated the people who worked for the $President_1$.

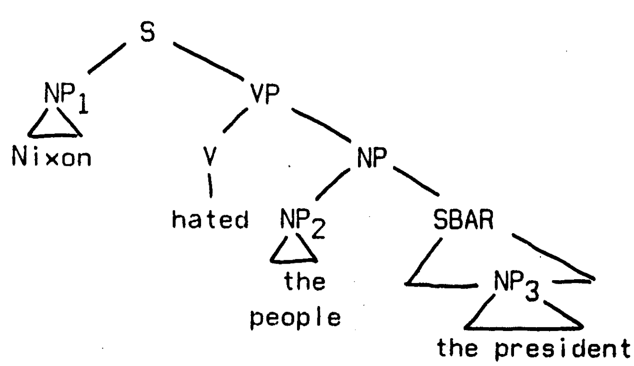
(129b) $Nixon_1$ hated the people who worked for him_1.

(129c) The people who worked for $Nixon_1$ hated the $President_1$.

(129d) The people who worked for him_1 hated $Nixon_1$.

The indicated coreference of sentence (a) is ruled out since it has the surface structure

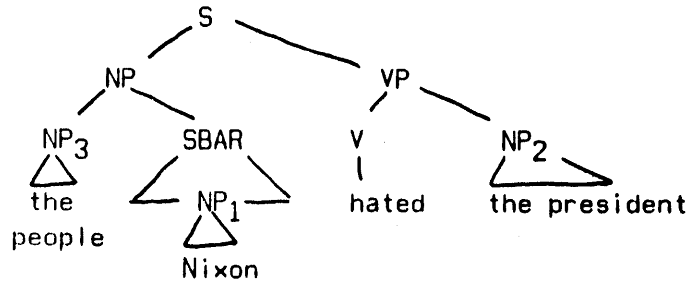
(130)



The first branching node above NP1, S, dominates NP3. Hence, NP1 c-commands NP3.

Since NP3 is not a pronoun, the Non-coreference Rule blocks coreference. When NP3 is a pronoun, as in (b), the coreference is not blocked. Sentence (c), on the other hand, has the surface structure

(131)



Here, neither NP1 nor NP2 c-command anything. Hence, coreference is free. It is even possible to use a pronoun for NP1, as in (d) -- a counter-intuitive phenomena which has fascinated linguists for years.

The Non-definite Anaphora rule was originally motivated by a desire to limit backwards pronominalization, such as in (d), to cases where the antecedent was a definite np:

(132a)

* The people who worked for him_1 hated { a $president_1$. / each $president_1$. / no $president_1$. }

(132b)

* The people who worked for { a $president_1$ / each $president_1$ / no $president_1$ } hated him_1.

If the antecedent has a certain form, it must c-command the pronoun in order to corefer with it. In (a), the lowest branching node above the antecedent is the VP. Hence, the antecedent c-commands neither the subject nor the pronoun inside the subject. In (b), the lowest branching node above the antecedent is the *for* pp. So the antecedent doesn't c-command the pronoun here, either.

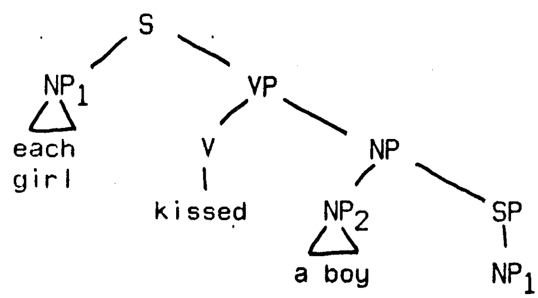
On the other hand, coreference is not blocked in the following

(133). A $president_1$
Each $president_1$ } hated the people who worked for him_1.
No $president_1$

since the subject c-commands everything in the verb phrase.[1] In order to extend these rules to quantifier scope, the term "corefer" is replaced by the term "coindex", and "pronoun" is replaced by "nps without descriptive content". The latter stipulation is necessary in order to deactivate the Non-coreference rule, which would otherwise rule out the different/per reading of

(134a) Each girl kissed a boy.

(134b)



Since *each girl* c-commands the Skolem modifier, the Non-coreference rule will block coindexing unless Skolem modifiers are, like pronouns, explicit exceptions to the rule. But this is not an unreasonable stipulation. Lasnik 1975 points out that epithets behave like pronouns with respect to the Non-coreference rule. Thus,

(135) $Nixon_1$ hated the people who worked for the $bastard_1$.

is acceptable. Hence, the replacement of "pronoun" with "nps without descriptive content" is independently motivated.

The reformulated rules are

1. Indefinite subjects sound odd unless placed in an appropriate discourse context -- eg. the first line of a fairy tale.

(136a) Non-coindexing
If X and Y are nps such that
X c-commands Y in surface structure, and
Y has no descriptive content,
then X and Y can not be coindexed in logical form.

(136b) Non-definite Coindexing
If X and Y are nps such that
X is a non-definite np, and
X doesn't c-command Y in surface structure,
then X and Y can not be coindexed in logical form.

With these rules, the clausal extreme of the embedding hierarchy is predicted. When the distributive np is embedded, as in

(137a) 0% Yesterday at the conference, I managed to talk to a guy who is representing each raw rubber producer in Brazil.

(137b) 7% That each aide knew about the hush money was proved with a secretly taped conversation.

The non-definite nps, ***each producer*** and ***each aide***, do not c-command the nonspecific nps ***a guy*** and ***a secretly taped conversation***. Consequently, they don't c-command such Skolem modifiers as the nonspecific nps might have. The Non-definite Coindexing rule applies, blocking coindexing. Hence, very few informants should report a different/per interpretation, which is in fact the case.

When a nonspecific np is embedded, as in

(138a) 66% Striking airline workers forced several major airlines to cancel every flight which was going to an eastern airport today.

(138b) 55% Each secretary reminded me to schedule an appointment.

(138c) 16% Each secretary reminded me that I should schedule an appointment.

the non-definite nps, ***every flight*** and ***each secretary***, c-command the indefinite nps, ***an eastern airport*** and ***an appointment***, and hence their Skolem np modifiers as well. Thus, the Non-definite Coindexing rule will not block coindexing. With coindexing free, the informants could be expected to report a mixture of per readings. And in fact they do.

The transformation theory predicted symmetric judgements for the embedding of ***each*** and of ***a(n)***. That is, the judgements on (138) should have been 100%, just as the judgements on (137) were 0%. This symmetric prediction is false. The anaphoric theory, on the other hand, predicts ambiguous readings for embedded ***a(n)***, and unambiguous same/per readings for embedded ***each***. This prediction fits the data somewhat better.

4.6 Clausemates and C-command

When clausemate nps are considered, the evidence is less decisive. As pointed out in section 2.5, c-command and surface order are equally poor predictors of quantifier scope. But on the other hand, c-command is also a poor predictor of coreference when both nps are in the verb phrase (See Reinhart 1975, sections 4.3 and 4.5). This indicates that c-command might be the wrong structural predicate for describing these phenomena, but it does not invalidate the anaphoric theory of quantifier scope. The defense of the anaphoric theory requires only that blockages of the different/per relation be found wherever blockages of coreference occur. As example, take

(139a)	59%	Each boy is kissed by Rosa in a picture of mine.
(139b)		Each kid_1 gets kissed by Rosa in his_1 picture.
(140a)	0%	Rosa kisses each boy in a picture of mine.
(140b)	*	Rosa kisses each kid_1 in his_1 picture.
(141a)	100%	Rosa put each book in an envelope.
(141b)		Rosa put each $book_1$ in its_1 envelope.
(142a)	100%	Each book was put in an envelope by Rosa.
(142b)		Each $book_1$ was put in its_1 envelope by Rosa.

The logical forms of (a) and (b) are isomorphic: the indefinite nps of (a) have Skolem modifiers just where the pronouns are in (b). As shown, coindexing is blocked only in (140), and there it is blocked for both quantifier scope and anaphora. In this fashion, anaphora and quantifier scope can be compared without making any assumptions about the c-command relations of ***kiss NP in a picture*** versus ***put NP in an envelope***.

Indeed, for some rather extreme quantifier scope examples, there are analogous anaphoric examples.

(143a) 100% We studied each two-car collision carefully. A driver who had been drinking was at fault.

(143b) We studied each two-car collision$_1$ carefully. One of its$_1$ drivers usually turned out to have been drunk.

Here, the Non-definite Coindexing rule has been violated, possibly because the ***each*** np is the topic of discourse. Crucially, both the different/per relation and the coreference relation are allowed to extend across sentences in this situation.[1] This shows that the constraints on them are quite similar.

There are certain cases when symmetry fails among the clausemates. One consistent source of asymmetry is the dative shift transformation. Dative shift creates a large difference in readings when the indirect object is non-specific, as in the following example:[2]

(145a) 70% Mary intends to mail each of her suicide notes to a friend.
(145b) 0% Mary intends to mail a friend each of her suicide notes.

But this large difference doesn't occur when it is the direct object that is nonspecific. There seems to be such an overwhelming preference for a different/per reading in this case that dative shift makes little difference, even when the articles are adjusted to favor the same/per reading:

(146a) 55% Mary intends to mail a couple of suicide notes to her friends.
(146b) 66% Mary intends to mail her friends a couple of suicide notes.

This subregularity can be captured in the rule

---

1. The different/per readings can not be a case of intersentential deletion of some modifier of ***a driver*** since the preceding sentence has been carefully constructed to lack an appropriate controller. This makes the ***it*** coreference a little harder to accept. A better example of intersentential non-definite anaphora is

(147) Each soldier$_1$ must run the course twice. He$_1$ must surmount all the obstacles without aid.

2. Incidently, this example is one of the many examples that show that a theory based on deep structure roles, such as direct object and indirect object, is empirically inadequate. See Ioup 1975 for such a theory. Note that she does not control for lexical content. Hence, her results may be interpreted as a cross-language correlation of pragmatic knowledge and deep structure roles.

(148) If X and Y are two nps such that
X is a dative-shifted direct object, and
Y is an argument of a dative-shifted indirect object, and
Y has no descriptive content,
then X and Y can not be coindexed in logical form.

In other words, if dative shift has occured, then one must be able to determine the referent of the indirect object independently of the direct object. The effects of this rule can be seen with anaphora, although the judgements are not as clear as one would like.

(149a) Mary intends to mail the $trophy_1$ to its_1 new home.
(149b) * Mary intends to mail its_1 new home the trophy.

(150a) Mary intends to mail Bob_1 his_1 trophy.
(150b) ? Mary intends to mail his_1 trophy to Bob_1.

The rule blocks coreference only in (149b). Consequently, coreference is much harder to get in (149b) than in (150b).[1] Thus, it seems that rule (148) is an appropriate way to account for the asymmetry of these examples. To write these rules in a theory based on predicate calculus, one would have to explicitly distinguish $\exists$ from $\forall$ -- an unmotivated increase in the descriptive power of the theory's translation rules.

4.7 Summary

The anaphoric theory has successfully accounted for the extremes of the embedding hierarchy, just as the transformational theory did. However, it also predicts the asymmetry of *each* embedding and *a(n)* embedding. Moreover, it predicts the c-command hierarchy among the clausemates, which May's theory can not do in a well motivated manner. This is probably the greatest empirical virtue of the anaphoric

---

1. Reinhart notes (See Reinhart 1976 section 4.2) that possessive pronouns are subject to a dialectal difference. In one dialect, possessive pronouns c-command only the np that they modify. For these people,

(152) His_1 students respect Ben_1.

is okay, and (149b) should be acceptable for them as well. In the other dialect, the possessive pronoun seems to c-command whatever the modified np c-commands. These speakers find coreference impossible in (152), and should find (150) unacceptable, too.

theory.

The anaphoric theory and the transformational theory appear equally well motivated. Whereas May had to postulate that S was the only bounding node of logical form, this theory must postulate the existence of "Skolem modifiers" and formal semantics for the resulting logical form.

There is a major problem that confronts the anaphoric theory. The structural predicate "c-command" is only a rough description of the syntactic constructions which block coindexing. There is no doubt that it is somewhat better than its predecessor "precede and command", but there are too many counterexamples and subregularities.

The worst irregularity, from a quantifier scope point of view, is alluded to by Reinhart in a crucial footnote. Consider the following quite ordinary quantifier scope sentence.

(153) For each possible answer, a formula is recorded in the data pool.

Sentences like this, with a preposed pp containing ***each***, usually have unambiguous different/per readings. However, the first branching node above the ***each*** np is the pp node. Hence, the np c-commands only the preposition ***for***. Since it doesn't c-command ***a formula***, coindexing should be blocked by the Non-definite Coindexing rule. Thus, the sentence is incorrectly predicted to have a same/per reading. Reinhart suggests calculating c-command with respect to the whole pp when the np is quantified. This turns out to be a very powerful idea. In the next section, it will be seen that this idea can be carried a little further, and replace c-command altogether.

5. *A Theory Based on Lexical Composition*

The embedding hierarchy and the distributive/collective hierarchy are the most well behaved correlations of quantifier scope and surface structure. The theory to be presented next, which is called IP raising (for "iteration phrase" raising), has been designed around these principles. Although the theory involves movement of quantified nps, just as May's theory did, it is motivated by a semantic phenomena, lexical composition, rather than constraints on syntactic transformations. Lexical composition is the name given to the process that builds the lexical content of a phrase from the lexical content of its constituents. Although very little is known about this process, one constraint that is widely accepted will be used to motivate the IP raising theory.

5.1 *Each Marks Iteration*

The basic idea of IP raising can be attributed to Theodore Vendler (Vendler 1967). He observed

> Suppose I show you a basket of apples and I tell you
>
> Take all of them.
>
> If you started to pick them one by one, I should be surprised. My offer was sweeping: you should take the apples, if possible, "en bloc." Had I said
>
> Take every one of them
>
> I should not care how you took them, provided you do not leave any behind. If I say
>
> Take each of them
>
> one feels the sentence is unfinished. Something like
>
> Take each of them and examine them in turn
>
> is expected. Thus I expect you to take them one after the other not missing any.

> The anticipated response to the first order squares nicely with the collective role of all we brought out in the previous section. The other two orders are both distributive, yet with a marked difference in emphasis: every stresses completeness or, rather, exhaustiveness; each, on the other hand, directs one's attention to the individuals as they appear, in some succession or other, one by one. Such an individual attention is not required in vain: you have to do something with each of them, one after the other.

To put Vendler's idea into computer jargon, the role of ***each*** is to mark the loop variable of some iteration. Because ***each*** marks ***apples*** as the loop variable, one gets an image of "taking" actions, one per apple.

As Vendler points out, the command "Take each apple" feels somewhat odd. But a command like "Weigh each apple" lacks this strangeness. The explanation is that the iteration interpretation is the marked interpretation -- if there is no pragmatic reason for the iterative interpretation, as opposed to the default, non-iterative interpretation, then the sentence is infelicitous. It misleads the hearer into thinking that the iteration is important. Compare the three commands:

(154a)	Take each apple.
(154b)	Weigh each apple.
(154c)	Take each apple, and examine it closely.

Each is felicitous in (b) and (c), but not in (a). In (b), there is a pragmatic reason to emphasize the iteration reading: both the iterative and non-iterative readings are plausible, but weighing the whole basketful and weighing each apple individually are so pragmatically distinct that is worthwhile to use ***each*** to distinguish them.

In (c), the discourse justifies the use of ***each***. Whereas (a) is infelicitous because there is no plausible reason to contrast the iterative and non-iterative readings, in (c) one sees that such a contrast becomes felicitous if used in the next clause. That is, because both interpretations of ***Examine X closely*** are pragmatically plausible and distinct, an ***each*** is felicitous, even if it appears in the preceding clause. Since well formed discourse often sets up a context before using it, Vendler reports that ***Take each apple*** sounds "incomplete" in isolation, rather than sounding infelicitous, as one would predict should one justify ***each*** solely on pragmatic grounds, ignoring discourse usages.

The purpose of ***each*** is to mark the loop variables of some iteration, but surface structure does not indicate what portion of the sentence's meaning is being iterated. A reasonable logical form would show which predicates are part of the ***each*** iteration. That is, in

(155) John asked Bill to weigh each apple

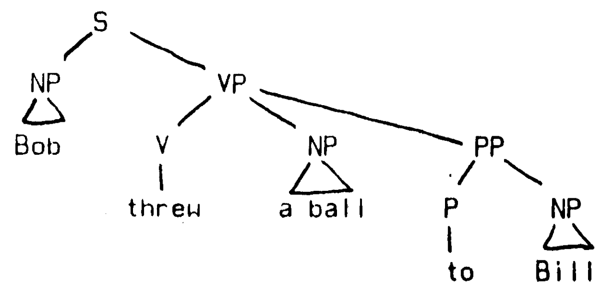
the predicate ***ask*** is not iterated, but ***weigh*** is. The logical form to be presented embodies a strong claim, but a well motivated one, about the relationship between surface structure, and the extent of an ***each***'s iteration.

To represent the idea of "doing something", a new lexical property, "iterability", will be placed on predicates in proportion to how pragmatically distinct their iterative interpretation is from their non-iterative interpretation. Thus, ***Weigh each apple*** sounds better than ***Take each apple*** because ***weigh*** is has more iterability than ***take***. A predicate's iterability, like all lexical content, is highly influenced by context.

5.2 IP Form

The procedure for translation into logical form can be motivated by a general principle of lexical content, called Strict Compositionality (Partee 1975). Strict Compositionality is a constraint on the translation of the lexical content of a sentence into the logical form. It states that the lexical content of a node can only compose with the lexical content of the node that immediately dominates it in the syntax tree. Thus, in the sentence, (156a) Bob threw a ball to Bill

(156b)



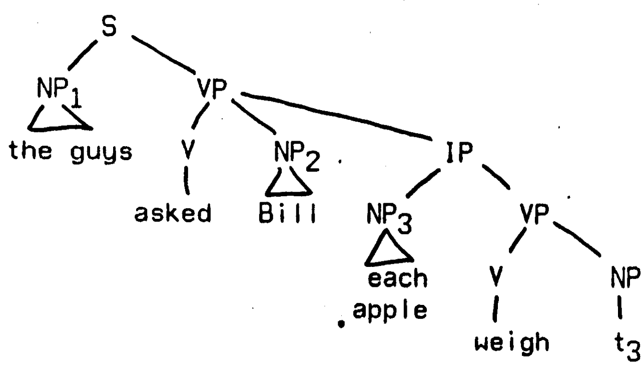
the np ***a ball*** can not compose directly with the pp ***to Bill***, yielding perhaps some kind of ball-vector. Instead, it must first compose with the verb phrase, yielding a ball-throwing action, which can then be combined with the pp to yield a ball-throwing-to-Bill action. To put it more graphically, lexical composition can occur only along the lines of the syntax tree.

Since the translation of ***each*** into logical form is constrained by Strict Compositionality, the extent of its interaction must correspond to some constituent of surface structure that includes the ***each***. To see why, picture the process of repeated lexical compositions gradually moving material up the tree. The semantic marker for ***each***, so to speak, can only move up through the nodes dominating it, not across. The only lexical material that can interact with the ***each*** marker also must moves up through the nodes, and so its semantic markers can only collide with the ***each*** marker at some node that dominates them both. Thus, if the ***each*** marker stops rising at some node, only the lexical material beneath that node can interact with the ***each***. Hence, the extent of the iteration corresponds to the constituent dominated by the highest node that the ***each*** marker has risen to.

With this motivation, the logical form for IP raising can be presented. As in May's theory, it is surface structure that has been modified by removing an np, leaving a trace, and attaching it higher in the tree. However, instead of Chomsky-adjoining the np to S, a new node, IP (for iteration phrase), is created, and the np is daughter-adjoined to it. As an example, take

(157a) The guys asked Bill to weigh each apple.

(157b)



Since IP dominates only the lower verb phrase, the predicate ***weigh*** is part of the iteration but the predicate ***ask*** is not. As in May's theory, traces are coindexed with the moved nps.

The representation of specificity and definiteness will be represented as binary features on the nps. Thus, the np ***the guys*** is +specific and +definite. These features will be left out of the following illustrations unless they are important.

On the other hand, distributivity will not be represented as a feature. Instead, the following stipulation will be made:

(158) An np is distributive with respect to a predicate P if and only if it is an argument (i.e. daughter) of some IP that dominates P.

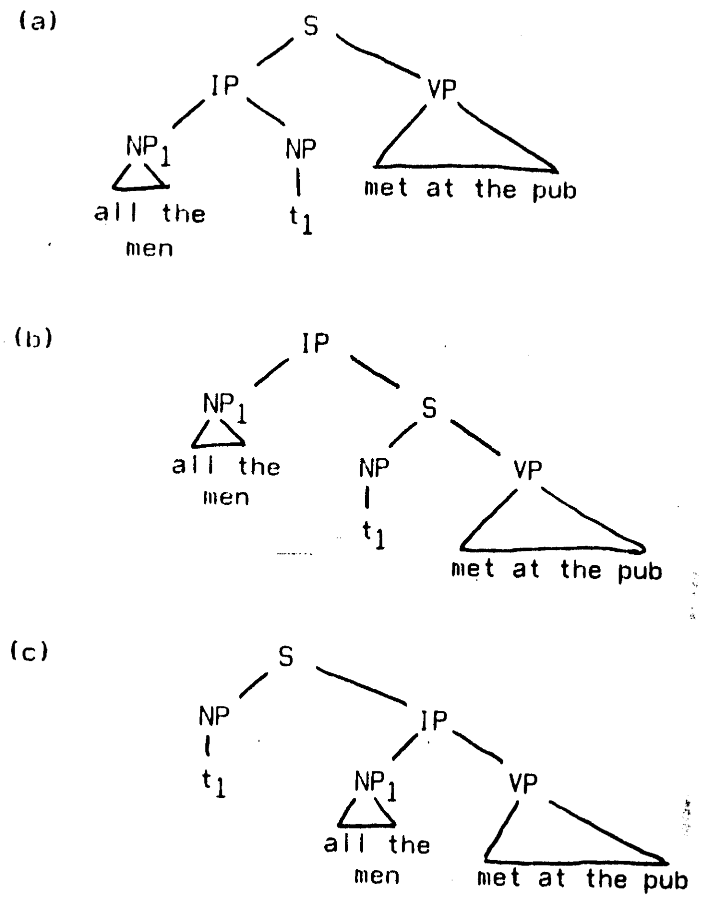
Thus, the np ***each apple*** is distributive with respect to ***weigh*** and the np ***the guys*** is collective. However, ***each apple*** is collective with respect to ***ask***. That is, the guys didn't point to each apple in turn, saying "Bill, please weigh that, that, that, and that." Instead, they asked that the collection of apples be weighed, and that they be weighed individually.

There is only one well-formedness constraint on this logical form, which will henceforth be called IP form. That constraint is the familiar Condition on Proper Binding. It states that the moved np must c-command its trace (See Section 3.1 for an accurate statement). This implies that the IP of a distributive np must dominate the np's trace. Figure 14 illustrates this constraint.

(c) is ill-formed since ***all the men*** doesn't c-command its trace. (b) is well-formed

Fig. 14. The Condition on Proper Binding and IP Form

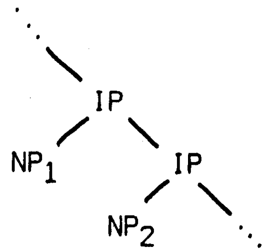
All the men met at the pub.



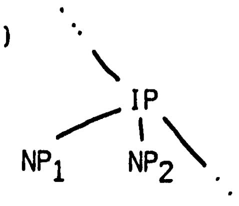
but nonsensical, since it makes ***all the men*** distributive with respect to ***meet***, and ***meet*** has a selectional restriction that requires a set of men as its subject whenever it is intransitive. (a) is well-formed and sensible, since ***all the men*** is now collective with respect to ***meet***.

IP form is quite similar to the typed predicate calculus that May uses. However, in May's logical form, quantifiers are adjoined only to S -- in IP form, they can be adjoined at any level. In May's logical form, quantifiers must nest -- here they can either nest, as in (a) below, or be sisters as in (b).

(159a)



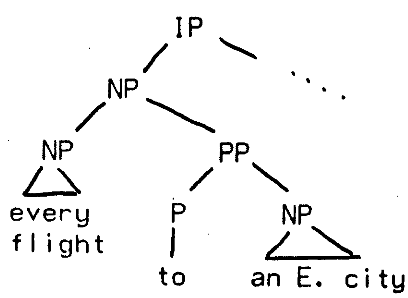
(159b)



(More on this distinction in a moment). Lastly, and most importantly, existential quantifiers are moved in May's theory but not in this theory.

The reader will recall that May's theory used movable existentials and the Condition on Proper Binding to account for indefinite nps in pp modifiers (eg. ***every flight to an eastern city***). In IP form, such constructions are represented (assuming ***every*** is distributive) as

(160)



Since the IP dominates the predicate ***eastern city***, one would expect it to be iterated. Hence, one would expect a different/per reading, which in fact does not occur. But such expectations would be based on an oversimplified notion of what this logical form <u>means</u>.

The formal semantics of IP form is given in the appendix. It is a straightforward combination of Woods' formal semantics for typed predicate calculus and Tarski's formal semantics. These two techniques work well together because both use the same control strategy, namely argument order evaluation. That is, the arguments of a predicate (or function) are extended before the predicate is. This is the familiar depth first evaluation which is the default control strategy of LISP, ALGOL, FORTRAN, and most other programing languages.

Argument order evaluation is responsible for the same/per reading of example (160). Moved nps are considered to be arguments of the IP. Hence, they are evaluated first, returning sets of objects -- their extensions. Next the IP iterates through the elements of these sets, repeatedly binding the appropriate traces to elements of the sets (This is just like the multiple-variable DO loop of MACLISP, SAIL, and other programming languages). With these bindings, it evaluates the reminder of the logical form that it dominates.

What this all means in terms of per relations is this: When a non-specific np is an argument of an IP, it is evaluated before the IP, so it is unaffected by the iteration. Similarly, if it is an argument of an argument, etc. of an IP, it is unaffected by the iteration. The whole of this discussion can be summed up in the following statement of what the logical form means:

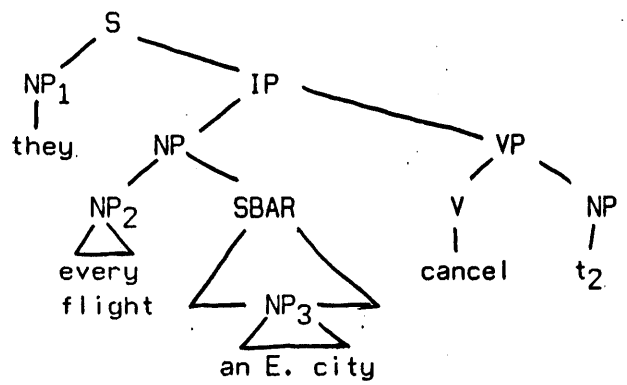
(161a) If NP1 is a nonspecific np, and
    IP1 is an iteration phrase with NP2 as an argument, and
    IP1 dominates NP1, and
    NP1 is not a *argument of IP1,
  then the informant will report a different/per relation.

(161b) <u>*argument</u>
  X is a *argument of Y if and only if
    X is an argument of Y, or
    X is an argument is Z, and Z is a *argument of Y.

In other words, ***every flight to an eastern city*** has a same/per reading because ***an eastern city*** is an argument of ***flight***. Relative clauses are not arguments. Hence, the familiar example

(162a) They canceled every flight which was going to an eastern city.

(162b)



has a different/per reading, when ***an eastern city*** is nonspecific, because ***an eastern city*** is dominated by the IP but is not a *argument of it.

The two extremes of the ***a(n)*** embedding hierarchy are thus accounted for by stipulating that possessive and pp modifiers are arguments of nps, but relative clauses are not. As with the two preceding theories, there is no explanation for why the reduced relative is halfway between the pp and relative clause modifiers.

When the embedding construction is not a modifier, it can't be an argument. Hence, all versions of

(163) Each secretary reminded me { about the scheduling of an appt. / about scheduling an appointment. / to schedule an appointment. / that I should schedule an appt. }

are equally open to a different/per reading. This agrees with the data -- all four versions of (163) have about the same degree of ambiguity.

There is one direct argument for IP form. But it is an adequacy of expressive power argument, similar to the Hintikka argument for Skolem form. IP form is able to represent certain sentences that are difficult for predicate calculus to represent. Consider

(164a) Each sequence was given a rating by each subject.

(164b) Each cork is carefully fastened to each Champagne bottle with a prefabricated wire basket.

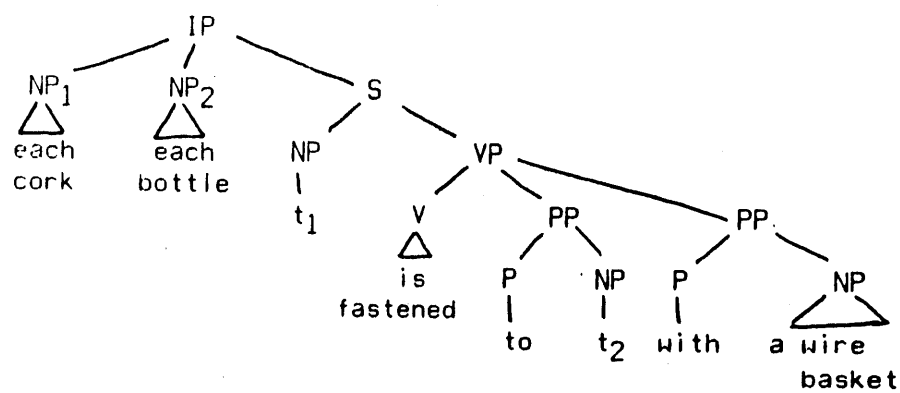
Predicate calculus would represent (a) as

(165) $\forall x\ [\text{sequence}(x) \supset \forall y\ [\text{subject}(y) \supset \exists z\ [\text{rating}(z)\ \&\ \text{give}(y\ x\ z)\ ]]]$

That is, for all possible subject-sequence pairs, there was a rating given. If there were 3 sequences and 7 subjects, there would be 21 pairs, and hence 21 different ratings.

But in (b), 85% of the informants claimed that the corks and bottles are paired one to one. If there were 10 bottles and 10 corks, then there would be only 10 pairs, not 100. hence, there would be just 10 wire baskets -- one per cork/bottle pair. IP raising can represent this reading by associating both moved nps with the IP. That is,

(166)



The formal semantics of this logical form expression turn out to be a loop with two loop variables, whereas the semantics of (164a) is two loops, one nested inside the other.

Predicate calculus can represent (164b), of course. However, one of the ***each*** nps would have to be given a collective, non-specific reading. This would result in the expression

(167) $\forall x\ [cork(x) \supset \exists y\ [bottle(y)\ \&\ \exists z\ [basket(z)\ \&\ fasten(x\ y\ z)]]]$

But such an interpretation of ***each*** violates two of the strongest article interpretation rules, namely that definite nps are specific and that ***each*** is distributive. What one might prefer is to augment the logic with a $\forall\forall$ operator that binds a pair of variables:

(168) $\forall\forall x,y\ [[cork(x)\ \&\ bottle(y)] \supset \exists z\ [basket(z)\ \&\ fasten(x\ y\ z)]]$

Skolem form could be similarly augmented to accommodate (164b) while avoiding violence to the article interpretation rules.

This argument is another argument based on inherent inadequacy of expressive power. As with the Hintikka argument, it show that predicate calculus must be enriched, or the article/quantifier map must be changed. On the other hand, IP raising can represent the troublesome reading with ease.

5.3 Translation Rules

Having discussed the logical form and its expressive power, it is time to examine the rules for translating into this logical form. As mentioned previously, the basic idea behind the translation is to raise the iteration phrase and its distributive np. In order to account for the embedding hierarchy, it is postulated that the final resting place is determined by the following two factors.

The factor driving the IP upward is due to the listener's "desire" to have a pragmatically felicitous iteration. A low resting place means only a small constituent is part of the iteration -- if that constituent doesn't make sense as an iteration, but a larger constituent would make sense, the listener tries to move the IP up.

Opposed to this desire for felicity is the second factor, the "effort" associated with moving the IP up one node. (Although I mean the terms "desire" and "effort" to be taken as an analogy to the rules' operation, and not as psychologically measurable quantities, I ask the reader to recall the main assumption of this report, namely that quantifier scope is disambiguated by a thoughtful analysis, AFTER the meaning of the sentence has been arrived at by real linguistic processes.)

An example will make the operation of these two factors clear. The paradigmatic example of the embedding hierarchy is repeated below in simplified form:

(169a) I talked to a representative of each producer.
(169b) I talked to a man representing each producer.
(169c) I talked to a man who is representing each producer.

Now, note that it is slightly odd to say

(170) Mortimer is representing each producer.

The use of ***each*** is infelicitous because there is little pragmatic contrast between being a representative for a group of producers, and repeatedly representing one producer at a time (if the later makes sense at all!). Hence, having only the concept representing-producers in the iteration of (169) would give the sentences a low iteration desirability. Because it will soon be necessary to compare magnitudes, let this low desirability be given a value, say +1.

On the other hand, the sharp pragmatic contrast of talking to a group of men, and buttonholing them each individually, makes the following use of ***each*** quite felicitous:

(171) Mortimer talked to each man.

It is highly desirable that the concept of talking-to-men be part of the iteration of (169). Let its desirability be +5.

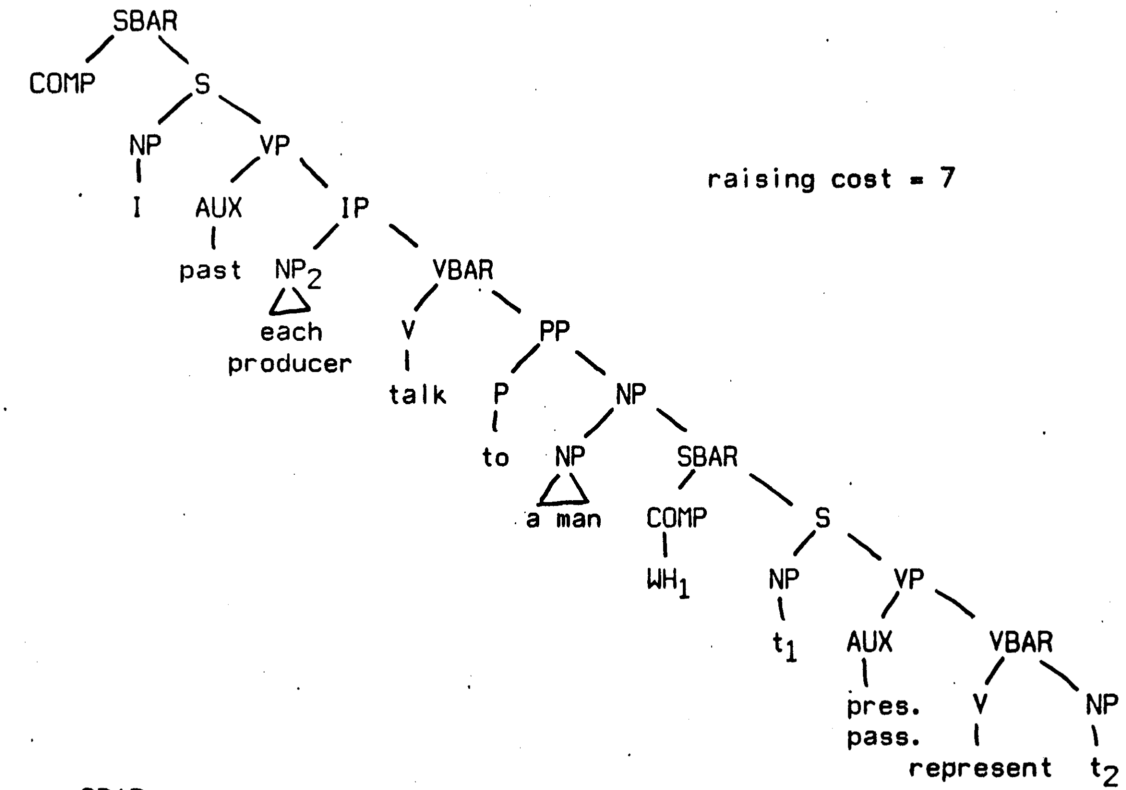
Now the effort associated with raising the IP to dominate ***talk*** is much greater with the full relative clause than with the PP construction because the ***each*** np is more deeply embedded -- the IP has further to climb. Let us deduct a "raising cost" of 1 for each node that the the IP must be raised. Using the surface structures of figures 15, 16 and 17, one can verify the cost-benefit analysis of figure 18.

Since the non-specific np is dominated by the IP only in the talk-to-men interpretation, the embedding hierarchy is now completely predictable. One postulates that

Fig. 15. Raising Cost of Full Relative Clause

I talked to a man who is representing each producer.

(a)



(b)

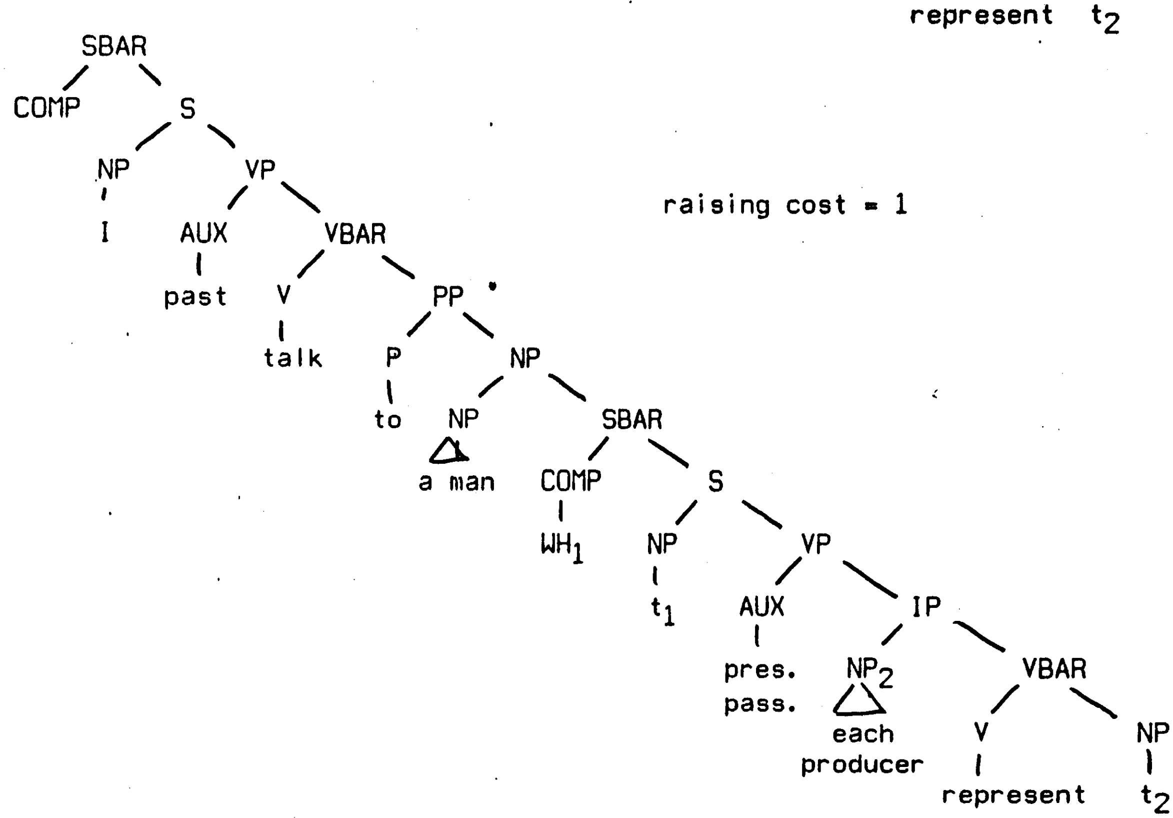
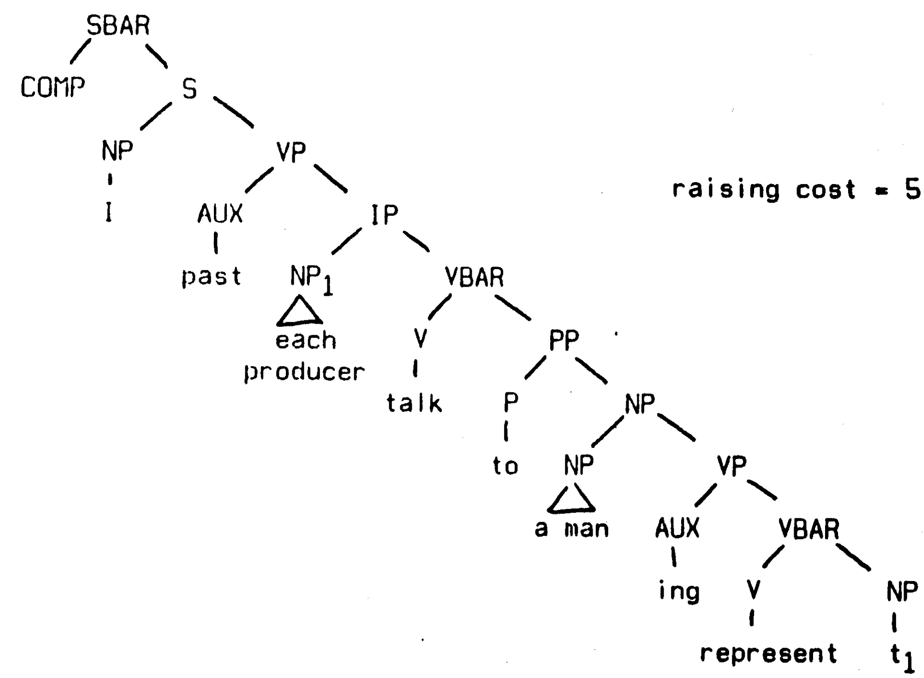


Fig. 16. Raising Cost of Reduced Relative Clause

I talked to a man representing each producer.

(a)



(b)

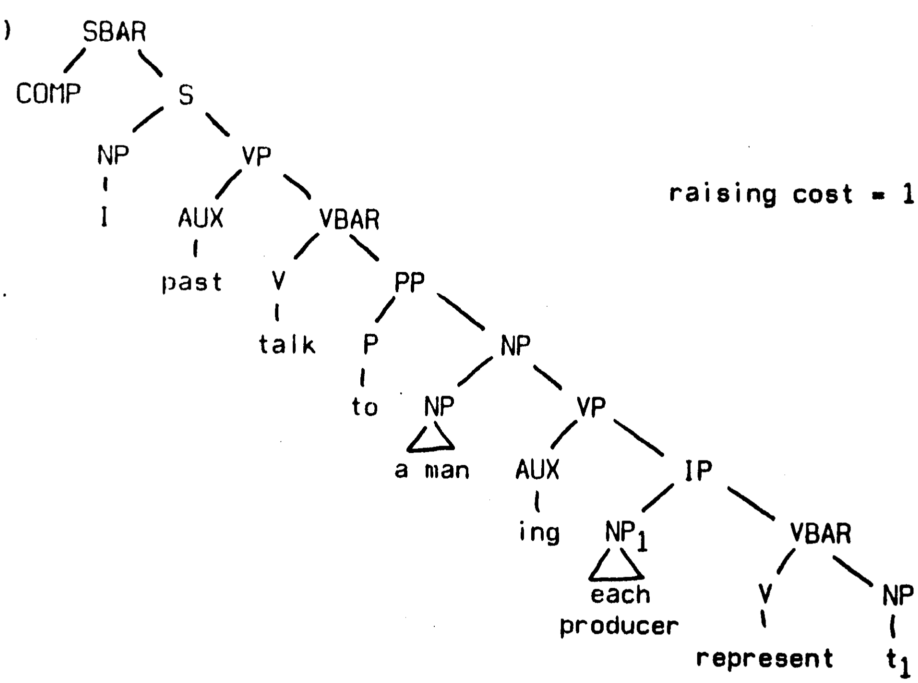
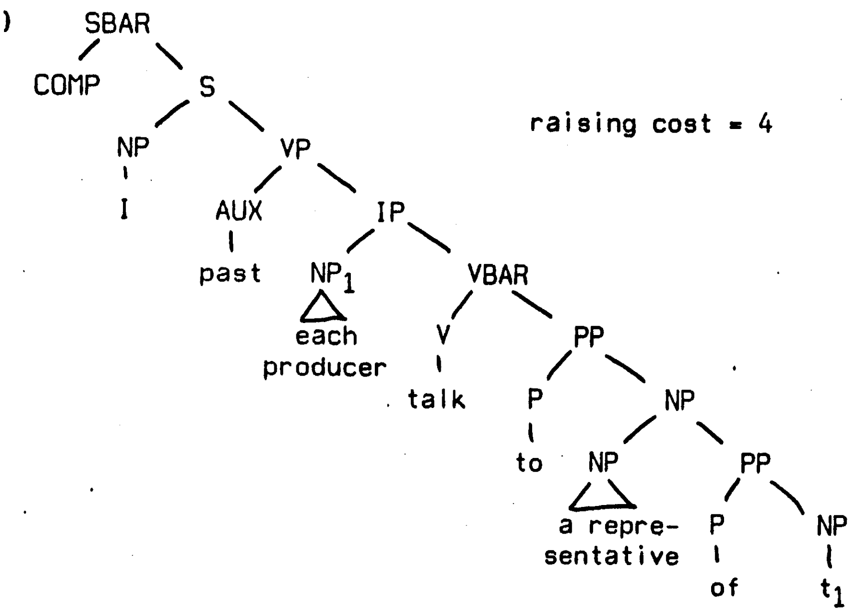


Fig. 17. Raising Cost of PP Modifier

I talked to a representative of each producer.

(a)



(b)

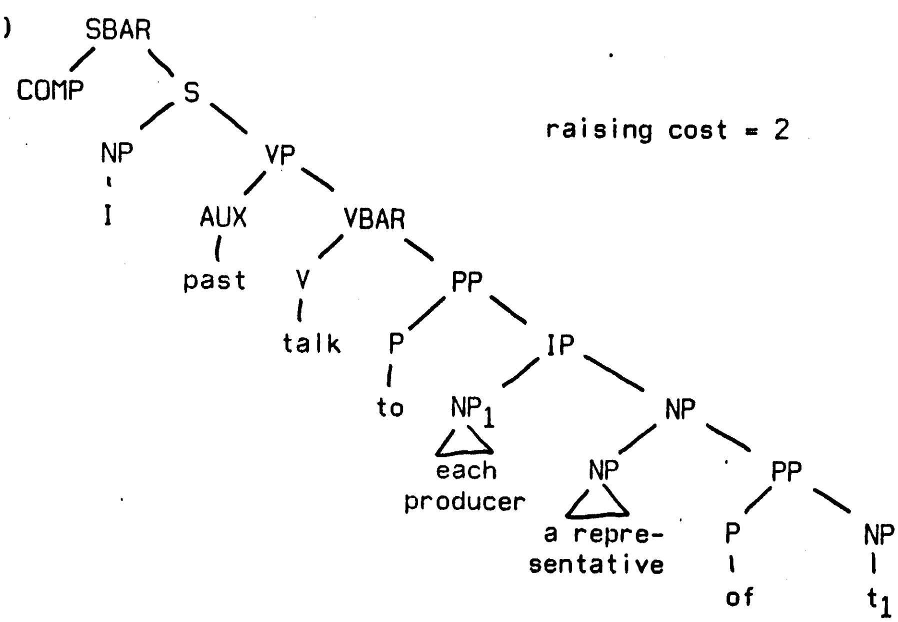


Fig. 18. An Analysis of the Embedding Paradigm

	iteration desirability	raising cost	total value
Full Rel. Clause			
Talk-to-men	+5	-7	-2
Represent-producer	+1	-1	0
Reduced Rel. Clause			
Talk-to-men	+5	-5	0
Represent-producer	+1	-1	0
PP modifier			
Talk-to-men	+5	-4	+1
Represent-producer	+1	-2	-1

(172) The prefered reading of a sentence is the reading with the highest total value, where total value is defined as the iteration's desirability minus the raising cost.

This rules predicts that the full relative clause will have the same/per reading, the reduced relative clause will be ambiguous, and the np-pp construction will have a different/per reading.

The prediction of ambiguity for the reduced relative clause is unique to the IP raising theory. The other two theories use two separate rules to translate embedded structures, one for subordinate clauses and one for subordinate pps. The ambiguity in those theories can be captured only by introducing a syntactic ambiguity in surface structure, or by writing a special rule for reduced relative clauses. It is clear that IP raising is much simpler.

There is empirical evidence which gives direct support to IP raising. As mentioned above, if the extent of the iteration includes only predicates of iterability, the *each* sounds infelicitous. That is,

(173) *each* is felicitous as an article for an np only when, in the prefered interpretation of the sentence, a predicate of high iterability is dominated by an IP which has that np as an argument.

Crucially, this predicts that the judgment of infelicity is correlated with the iterative desirability of figure 18, not the total value.

In particular, it predicts that the full relative clause, where ***talk*** is not dominated by the IP, will sound infelicitous. Indeed, two thirds of the informants who read the full relative clause sentence said, without being asked, that it sounded odd, and that they would use ***every*** or ***all*** instead of ***each***.

Even more importantly, all the informants who got an unambiguous same/per reading on the reduced relative clause sentence commented spontaneously that the ***each*** sounded bad. When asked to rephrase the sentence, they replaced the ***each*** with ***all*** or ***every***. On the other hand, the informants who gave the reduced relative clause a different/per interpretation had no complaints about the sentence.

These data show that it is not a low total value that makes a sentence sound odd, but a low iterability.

IP raising can account for part of the QR/WH interaction problem that plagued May's theory, namely the difference between embedded and non-embedded WH questions. One of the examples is repeated here:

(174a) Which city has each of the burglars been assigned to?
(174b) Woodward wanted Bernstein to find out which city each of the burglars had been assigned to.

Suppose that find-out is a highly iterable action, but assigning-burglar-to-city isn't. Suppose further that in some dialects, WH nps have low iterability, while in other dialects, they have a moderate iterability. This last assumption accounts for the dialect split.

In interpreting (a), informants with the low iterability WH dialect will have little desire to move the IP past assign-to. Those who stop the IP at assign-to will report a same/per reading, because the WH would not be dominated by the IP. Those who move the IP up to the SBAR node that dominates the WH will get a different/per reading. However, everyone with this dialect should complain about the sentence, because there is no predicate of high enough iterability to warrant the use of ***each***. But when presented with (b), these informants would associate the IP with find-out, which is a highly iterable action. Now the WH would be dominated by the marker, and the informants should report a different/per reading. There should be no complaints about the sentence. Thus, the low iterability WH dialect accounts for the informants who though (a) was bad and (b) was good.

Informants with the moderate iterability WH dialect will move the IP up to dominate the WH in (a). They should have no complaints about the sentence, and should report a different/per reading. Like the informants with the other dialect, these informants will interpret (b) by associating the IP with the highly iterable find-out, since WH is only moderately iterable. In both (a) and (b), readers with this dialect will get a different/per reading and have no complaints about the sentence.

As noted in section 3, two dialects with just these characteristics have been observed. Thus, IP raising gives one just the right power to explain the interaction of WH and quantifiers.

5.4 Conclusions

Unfortunately, it is inaccurate to make the raising cost proportional to number of nodes between *each* and the IP. To get simple proportionality to correctly predict the readings of the subject nominalizations of figure 5, some nodes must be pruned from the surface structures of the bare gerund and bare infinitive. Such pruning could probably be motivated on syntactic grounds. However, a more serious problem occurs when embedding structures are nested one inside the other. Consider, for example,

(175) I talked to a representative of the manufacturer of the critical part of each design.

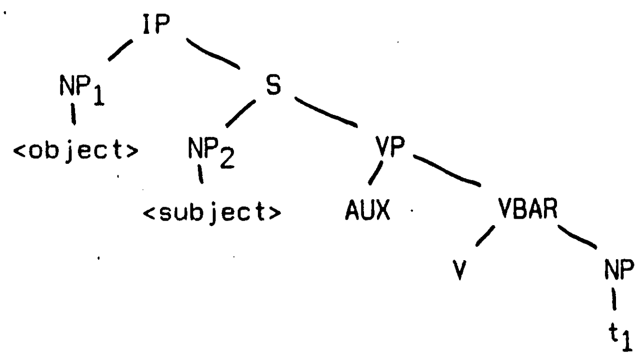
To my intuition, this sentence entails a different representative per design. However, the IP and the *each* are separated by six nodes, which is more than the distance separting them in the case of the full relative clause. I suspect that a better correlation would require assignment of embedding cost with respect to the category of the embedding construction, rather than the depth of the *each*. However, such a revision would require a tabular assignment of costs, thus robbing the IP raising theory of the elegance which is one of its major attractions.

Like the anaphoric theory, IP raising can predict the c-command hierarchy,

(176) preposed pp > subject > sentential pp > verb phrase pp > object

For example, if the object is distributive, and the subject is nonspecific, then a different/per reading would require that the object be raised three nodes.

(177)



This is unlikely, unless the subject contains material of high iterability. However, when the syntactic positions of the two nps are reversed, say by passivization, then the distributive need not be raised at all -- if the indefinite is nonspecific, the sentence will have a different/per reading.

The kind of clausemate data that would distinguish the lexical theory from the anaphoric one would involve subregularities, such as the dative shift one:

(178a) 70% Mary intends to mail each of her suicide notes to a friend.
(178b) 0% Mary intends to mail a friend each of her suicide notes.

(179a) 55% Mary intends to mail a couple of suicide notes to her friends.
(179b) 66% Mary intends to mail her friends a couple of suicide notes.

It was noted in section 4.6 that the coreference facts seem to follow this pattern as well, thus providing an independent motivation for a rule describing this subregularity. However, I know of no lexical composition facts that could motivate such a rule, eg. that the shifted indirect object -- ***a friend*** in (178b) -- is an argument of the direct object.

Another interestinging subregularity involved inter-sentential different/per relations, eg.

(180) We studied each two-car collision carefully. A driver who had been drinking was at fault.

Again, lexical composition is too underdeveloped to allow independent motivation of a rule to cover this data. However, it may be that such a rule would expain that felicity of Vendler's example,

(181) Take each apple. Examine it carefully.

Presumably, the ***each*** is felicitous since ***examine*** has a high iterability. But the IP must dominate both ***each*** and ***examine***, which is impossible since they are in different sentences. Perhaps research in lexical composition will motivate a remote structure that spans more than one sentence.

In short, IP raising does predict the major clausemate correlations even though not enough is known about lexical composition to motivate the subregularities.

In summary, the lexical theory explains the asymmetry of embedding and the whole of both embedding hierarchies -- except reduced relative clauses containing indefinite nps. It also predicts the c-command hierarchy for clausemates. Thus, it is a viable competitor with the anaphoric and transformation theories. The greatest virtue of IP raising lies in its interaction with lexical content. In particular, no other theory has a mechanism that can account for Vendler's observations. The rule that predicts infelicity when the IP fails to dominate lexical material of high iterability is a unique contribution of this theory.

6. *Inconclusions and Speculations*

The major assumption of this report is that disambiguation of quantifier scopes is not a linguistic process. Instead, the correlations of quantifier scope judgments with aspects of surface structure are epiphenomena of real linguistic processes. This assumption explains, in an informal way, why lexical content is so much more influential than syntactic structure, and why the informants have so much trouble making a judgment when syntactic and lexical factors are opposed. All the informants have commented, at one time or another, that the sentences given to them didn't provide enough information to answer my questions -- they had to imagine a likely situation that the sentence could be describing, and answer my questions with respect to that particular situation.

Nonetheless, even in such "forced" data one sees an interesting correlation with grammatical forms. This report has concentrated on finding out which real linguistic processes might be causing these correlations. Three well known processes were investigated -- transformations, anaphora and lexical composition -- by using their known syntactic correlates to motivate theories of quantifier scope. All three of the resulting theories where able to account for the extremes of the major correlations.

The anaphoric theory is somewhat better motivated than the other two. No syntactic movements obey exactly the same rules as QR, and too little is known about lexical composition to motivate the details of IP raising. But the correspondence of Non-definite anaphora and quantifier scope is uncanny. Not only the major correlations, but some of the stranger minor correlations of quantifier scope are echoed in Non-definite anaphora. There is also an intriguing form of bound quantifier scope, analogous to the reflexive pronouns.

Unfortunately, the treatment of embedding by the anaphoric theory is inaccurate. The c-command predicate just doesn't capture the interesting blend of readings that characterizes the *each* embedding hierarchy. Replacing c-command with a form of constituent movement (inspired by QR) led to the IP raising theory. This theory is the best predictor of the quantifier scope correlations. In addition, it explains an important class of unacceptable sentences, such as

(182) Yesterday at the conference, I managed to talk to the guy who is representing each raw rubber producer in Brazil.
[4] Weird sentence, the each sounds funny.

Unfortunately, attempts to combine the IP raising and anaphoric theory have been frustrated by an inability to show that Non-definite anaphora is bounded by IP raising. In fact, I am unable to reproduce some of Reinhart's results (most of the coreference judgements of section 4 are from Reinhart 1976). I suspect that my flashcard technique -- where informants read a sentence typed on a file card and paraphrase it -- is substantially different than whatever technique Reinhart used to collect her data. The difference in presentation technique may explain the difficulties I have had in verifying that Non-definite anaphora is bounded by IP raising.

So, the search for real linguistic processes to explain the quantifier scope correlations ends somewhat inconclusively.

6.1 Practical Suggestions

Several natural language engineers have asked me how quantifier scope should be handled, even if it must be handled in an ad hoc manner. The following is a speculative answer.

The basic framework should be the IP raising theory because it has the cleanest interface with deep, lexico-pragmatic information. There are three places where such information is needed. One is to determine the iterability of the predicates.

Another is to determine whether pp and reduced relative clause modifiers should be translated as type function arguments, or as class restrictions (ie. like possessive nps, or like full relative clauses). This information is necessary to disambiguate sentences like

(183a) Every man with a layered haircut is important.
(183b) Every stage of a layered haircut is important.

In (a), the pp modifier must be translated like a full relative clause so that the rules will allow the different/per reading, which it has. But in (b), the same pp must be translated as a type function modifier so that the two nps will have a same/per reading.

The third place that deep information is needed is to determine whether an indefinite np is specific or nonspecific. This seems to be the most difficult judgment to make. It may be the one which is causing the informants to balk. In general, if there is an obvious pragmatic relationship between the *each* np, the iterated predicate, and the indefinite np, then the different/per relation is called for, and one would give the indefinite np the nonspecific interpretation. If there is no obvious relation, then the sentence is very ambiguous. One would be quite justified in asking the speaker what s/he meant.

There is, however, an exception to this third rule of thumb. A pragmatic relationship need not be obvious when the sentence is asserting that it exists. As far as I can see, this occurs only when the *each* np is part of the first np in the sentence. It is especially common with preposed pps.

Asking the lexico-pragmatic component to search for an "obvious pragmatic relationship" would probably be facilitated if one represented nonspecific nps as nps with Skolem modifiers. Thus, disambiguation of function/argument relations and the hypothesized "obvious pragmatic relationship" could use the same machinery.

I have not implemented these heuristics, nor do I have any intention to do so. By their very nature, they depend on a good representation of lexical content and of discourse focus. Until some progress is made on these problems, it is unlikely that any system will exceed the performance of LUNAR.

7. *References*

Chomsky, Noam Syntactic Structures, Mouton, The Haugue, 1957

Chomsky, Noam "Conditions on Rules of Grammar" Linguistic Analysis 2.4, 1976.

Fauconnier, Giles "Do Quantifiers Branch" Linguistic Inquiry 6.4, 1975

Frege, Gottlieb "Begriffsschrift: A Formal Language, Modeled upon that of Arithmetic, for Pure Thought" 1879, in Jean van Heigenoort Frege and Godel, Harvard University Press, 1970.

Hintikka, Jaakko "Quantifiers vs. Quantificational Theory" Linguistic Inquiry 5.2, 1974.

Ioup, Georgette "The Treatment of Quantifier Scope in a Transfomational Grammar" City University of New York PhD dissertation, 1975, available from IU Linguistics Club.

Keenan, Edward "The Functional Principle: A generalization of the notion 'subject of'" Chicago Linguistic Society 10, 1974

Keyser, S.J. and Postal, P. Beginning English Grammar, Harper and Row, 1976

Kroch, Anthony "The Semantics of Scope in English" MIT PhD dissertation, 1974, available from IU Linguistics Club.

Kuno, Susumo "Three Perspectives in the Functional Approach to Syntax" in Chicogo Linguistics Society 11, 1975.

Jackendoff, Ray. S. Semantic Interpretation in Generative Grammar, MIT Press, 1972.

Lasnik, Howard "Remarks on Coreference" Linguistic Analysis 2.1, 1976.

Levin, Beth "The Instrumental Problem in English", MIT MS dissertation, in preparation.

May, Robert C. "The Grammar of Quantification" MIT PhD dissertation, 1977, available from IU Linguistics Club.

Partee, Barbara Hall "Montague Grammar and Transformational Grammar" Linguistic Inquiry 6.2, 1976.

Reinhart, Tanya "The Syntactic Domain of Anaphora" MIT PhD dissertation, 1976, available from IU Linguistics Club.

Ross, John Robert "A Nouniness Squish" MIT mimeo, circa 1974.

Smith, Brian "A Semantical Framework for Knowledge Representation" MIT MS dissertation, 1978.

Stockwell, R.P, Schachter, P. and Partee B.H. The Major Syntactic Structures of English, Holt Rinehart & Winston, 1973.

Vendler, Zeno "Each and Every, Any and All" in Vendler Linguistics in Philosophy, Cornell University Press, 1967.

Woods, William A. "Semantics and Quantification in Natural Language Question Answering" Bolt Beranek and Newman report 3687, 1977.

Woods, William A. "What's in a Link?" in Bobrow & Collins, eds. Representation and Understanding, Academic Press, 1975

Woods, William A., Kaplan, R. and Nash-Webber, B. "The Lunar Sciences natural Language Information System: final report" BBN Report No. 2378, 1972

8. *Appendix on Formal Semantics*

Several logical forms have been used that are new. The reader may be uncertain about what expressions in these logical forms mean. In order to reduce misunderstandings, this appendix gives a formal semantics for each of the new logical forms. That is, by providing an algorithm that answers the question "Is a given expression true in the world modeled by a given data base," the reader's understanding of the meaning of these logical forms will hopefully be clarified. Note that I am not going to cover the formal semantics with enough precision to establish their model-theoretic consistency.

The particular constructions to be covered are the nonstandard operators of typed predicate calculus, typed Skolem form, and IP form. In each case, an algorithm will be given which, given an expression and a data base, calculates whether the data base satisfies the expression. The algorithms and data structures will be written in pseudo-LISP. To make the code easier for non-LISP users to read, extra words have been added in lower case, and certain inessential constructions, such as PROG, have been omitted.

8.1 *Nonstandard Operators for Typed Predicate Calculus*

In section 2.2, it was argued that the three distinctions

(184) definite/indefinite
specific/nonspecific
collective/distributive

are best thought of as independent features. Thus, one needs eight quantifiers. Ordinary predicate calculus uses only two. $\forall$ is the definite, specific, distributive quantifier, and $\exists$ is the indefinite, nonspecific, distributive quantifier. This section gives formal semantics for the other six.

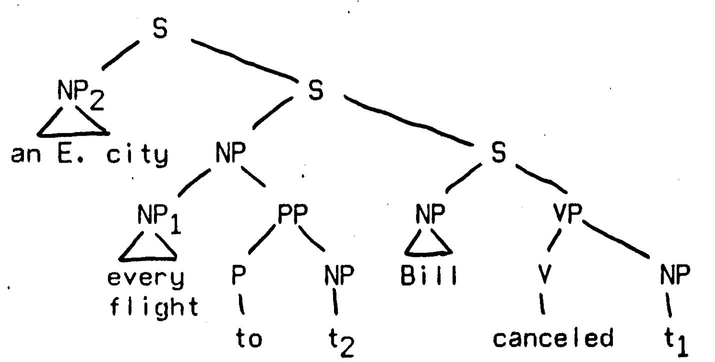
To make the algorithms precise, the LUNAR data structure for the typed predicate calculus will be used. Although the modified surface structure that May uses as logical form is quite perspicuous to humans, it has a great deal of information that is irrelevant. In particular, English predicate/argument relations are marked with pps, determiners, and surface order. It will be assumed that some conversion process, such as LUNAR's semantic interpretation rules, has disambiguated the predicate/argument relations, and stripped away the syntactic markings. The only substantive feature of this conversion process is

(185) If a nominal modifier has the form of a pp or possessive, then it is converted to an argument of the modified noun's type function. If the modifier is a reduced or full relative clause, it is converted to a class restriction on the noun's quantifier.

An example will make this rule, and the associated terminology, somewhat clearer.

(186a) Bill canceled every flight to an eastern city.

(186b)



(186c)

```
(FOR SOME X2 EASTERN-CITY () T
     (FOR EVERY X1 FLIGHT-TO (X2) T
          (CANCEL 'BILL X1)))
```

The quantifiers are nested in the same order in both the regular and the simplified logical forms. The three features, definiteness, specificity and distributive, are mapped into the appropriate one of the eight operators -- in this case, SOME and EVERY (ie. $\exists$ and $\forall$ respectively). The nps are everywhere converted to variables -- X1, X2, etc. Coindexed nps become the same variable.

Following the bound variable in the quantifier is the name of the type function -- EASTERN-CITY and FLIGHT-TO. Next, there are the arguments to the type function -- () and (X2) respectively. Note that the pp modifier of (b) has been converted to an argument, just as rule (185) says it should.

Following the type function's arguments, one would normally find the class restriction. The class restriction is a predicate of one argument which is intended to restrict the range of quantification to some subset of the extension of the type function. Since this example has no relative clauses, the class restrictions of both quantifiers are T, for true. Figure 19 shows an expression with a nontrivial class restriction. The complementizer is converted to the header of a lambda expression. This makes the proposition (FOR SOME ...(GO-TO ...)) into a predicate of one argument.[1]

As noted above, the control structure of Woods' formal semantics and LISP is the same -- argument order evaluation. Taking advantage of this, one can simply define FOR as a LISP function. This technique will be illustrated with the familiar quantifiers, SOME and EVERY. Figure 20 is the code for this simple version of FOR.

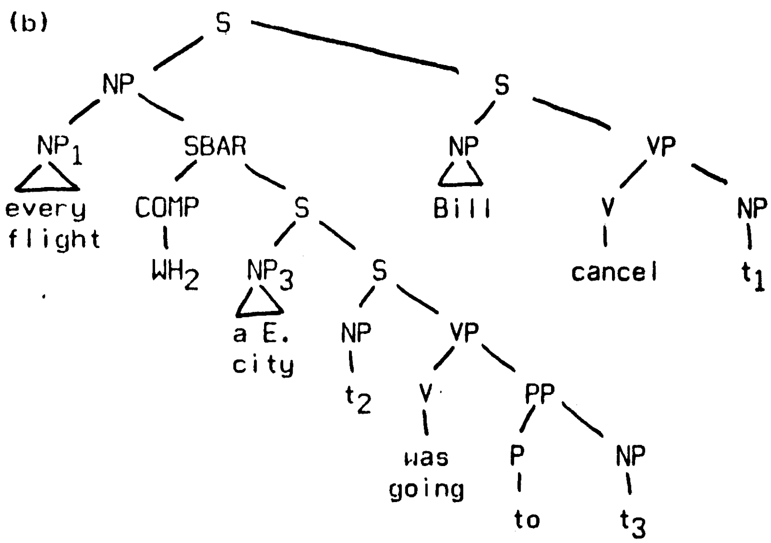
First FOR calls the type functions, then filters out the resulting set. This gives the

---

Fig. 19. A Nontrivial Class Restriction

(a) Bill cancelled every flight which was going to an eastern city.

(b)



(c)

```
(FOR EVERY X2 FLIGHT ()
              (LAMBDA (Y)
                      (FOR SOME X1 EASTERN-CITY () T
                           (GO-TO Y X1)))
      (CANCEL 'BILL X2))
```

---

1. This is one of several places where this syntax for the logical form differs slightly from LUNAR's.

Fig. 20. LUNAR's definition of FOR

```
(Define FOR (QUANT VAR TYPE/FUNCTION ARGS RESTRICTION EXPRESSION)
          RANGE ← (EXTEND/AND/FILTER TYPE/FUNCTION ARGS RESTRICT)
          SATISFIERS ← (SATISFIERS/OF/EXPRESSION RANGE VAR EXPRESSION)
          If QUANT=EVERY then
               (If RANGE=SATISIFIERS then (Return T) else (Return NIL))
          If QUANT=SOME then
               (If SATISIFIERS=NIL then (Return NIL) else (Return T)))

(Define EXTEND/AND/FILTER (TYPE/FUNCTION ARGS RESTRICTION)
          (Foreach X InTheSet (Apply TYPE/FUNCTION To ARGS) Do
               If (Apply RESTRICTION To X)=T then (Put X IntoTheSet RANGE))
          (Return RANGE))

(Define SATISFIERS/OF/EXPRESSION (RANGE VAR EXPRESSION)
          (Foreach X InTheSet RANGE Do
               (Substitute X For VAR in EXPRESSION)
               If (Evaluate EXPRESSION)=T then (Put X IntoTheSet SATISFIERS))
          (Return SATISIFIERS))
```

---

range of quantification. Next, FOR finds the values in range that satisfy the rest of the expression. If they all do, then both SOME and EVERY are true. If at least one does, then SOME is true but EVERY is false. If none do, then both SOME and EVERY are false.

This definition of FOR can be readily changed to implement the semantics of all eight quantifiers. As will be seen, the basic idea is to give a memory to each instance of an operator. The logical form will also be changed slightly; instead of QUANT being a symbol like SOME or EVERY, let is be a subset of the set {SPECIFIC DEFINITE DISTRIBUTIVE}.

The new definition of FOR is given in figure 21. The collective operators differ from the distributive ones in that they bind their variables to sets rather than to individuals. This is implemented by modifying the range. If the operator is collective, RANGE is replaced by the set of all possible subsets of itself, that is, by its powerset. For example,

Fig. 21. ***FOR for Nonstandard Operators***

```
(Define FOR (QUANT VAR TYPE/FUNCTION ARGS RESTRICTION EXPRESSION)

          RANGE ← (MEMORY VAR ARGS)
          If RANGE=CANT/REMEMBER then
               ( RANGE ← (EXTEND/AND/FILTER TYPE/FUNCTION ARGS RESTRICTION)
                    IfNot DISTRIBUTIVE ∈ QUANT then RANGE ← (Powerset RANGE))
          SATISFIERS ← (SATISIFIERS/OF/EXPRESSION RANGE VAR EXPRESSION)
          If SPECIFIC ∈ QUANT then
               (MEMORY VAR ARGS) ← SATISIFIERS
          If DEFINITE ∈ QUANT
               then (Return (RANGE=SATISFIERS))
               else (Return (Not (SATISFIERS=NIL))
```

(187a) Some flights collided.

(187b)
```
(FOR {SPECIFIC} X1 FLIGHTS () T
     (COLLIDE X1))
```

where ***flights*** is indefinite and collective, the powerset of all flights is the range. If one of those subsets is a set of planes that collided, then the sentence is true. If the example were

(188a) The flights collided.

(188b)
```
(FOR {SPECIFIC DEFINITE} X1 FLIGHTS () T
     (COLLIDE X1))
```

where ***flights*** is now definite, the sentence is true only if all subsets of ***flights*** are sets of smashed planes. Because of the pragmatic properties of COLLIDE, it would be sufficient to check only the largest subset of ***flights***, namely the set ***flights*** itself.[1]

The specific interpretation is distinguished from the nonspecific interpretation with the aid of MEMORY. This function, which would be tedious to define, simply remembers whatever it is told, indexed by its two arguments. If nothing has been stored for some particular values of its arguments, it returns CANT/REMEMBER. The basic idea of the specific reading is that its satisfiers must satisfy EXPRESSION regardless of the values of the free variables of EXPRESSION. So every time the specific FOR is called by some higher FOR, it remembers which values satisfied its EXPRESSION earlier, and checks only those for satisfaction this time. An example will clarify this implementation of the specific interpretation.

(189a) Every girl kissed a certain boy.

(189b)
```
(FOR {SPECIFIC DEFINITE DISTRIBUTIVE} X1 GIRL () T
     (FOR {SPECIFIC DISTRIBUTIVE} X2 BOY () T
          (KISS X1 X2)))
```

Since *a certain boy* is specific, the sentence is false in the following world:

(190)
```
(KISS LUCY CHARLES)
(KISS MARLA DON)
GIRLS = {LUCY MARLA}
BOYS = {CHARLES DON}
```

The first time the inner FOR is called, with X1=LUCY say, the RANGE must be calculated, and turns out to be {CHARLES DON}. Only CHARLES satisfies (KISS X1 X2), so the operator remembers {CHARLES} under the index (X2 NIL). The next time the operator is called, with X1=MARLA, MEMORY is called with the same index, namely

---

1. This property is very typical of collective predicates. In fact, I know of no counterexamples. Taking advantage of this, one could modify the definition of the collective reading to make it more efficient by replacing the line

```
RANGE ← (Powerset RANGE)
```

with the line

```
If DEFINITE ∈ QUANT
     then RANGE ← (MakeSingletonSet RANGE)
     else RANGE ← (Powerset RANGE)
```

This semantics for the definite, collective operator is somewhat more pleasing to the intuition.

(X2 NIL). So it returns {CHARLES}. But Marla did not kiss Charles, so SATISFIERS becomes NIL, the empty set. The operator returns NIL (ie. false). Thus, X1=MARLA doesn't satisfy the upper operator, and the whole expression is false.

If the sentence had a nonspecific object instead (eg. Each girl kissed a boy), nothing would be remembered about X2. So the inner operator would calculate {CHARLES DON} as the range each time it is called. Hence, there would always be satisfiers, and the inner operator would always return true. Hence, the sentence would be true.

If the sentence had been

(191a) Each girl kissed her boy.

```
(191b)    (FOR (SPECIFIC DEFINITE DISTRIBUTIVE) X1 GIRL () T
               (FOR (SPECIFIC DISTRIBUTIVE) X2 BOY (X1) T
                    (KISS X1 X2)))
```

where *girl* is an argument to *boy*, then the sentence is true in the world given above. Since MEMORY is indexed by the arguments of the np, {CHARLES} is stored under (X2 LUCY). Thus, when the inner operator is called the second time, MEMORY has nothing stored under (X2 MARLA). Hence the range is calculated rather than remembered, and the sentence turns out to be true.[1]

With these modifications, the FOR of Woods' semantics provides a formal semantics for all eight operators. The same techniques -- powerset and memory -- will be used to implement the semantics of the other logical forms.

8.2 Typed Skolem Form

The formal semantics for typed Skolem form will be implemented by translating into the typed predicate calculus described above.

In the preceding section, a conversion mapping translated the logical form with moved nps, traces etc. into a logical form with nested operators, predicates,

1. This definition of FOR assumes that RESTICTION have no free variables that might be bound by a higher FOR. Relaxing this assumption would require a third argument to MEMORY -- a list of the values of the RESTRICTION's free variables.

arguments etc. The converted logical form was then used as input to the formal semantics. A somewhat different conversion mapping will be used here.

The scope of quantifiers in typed Skolem form is represented in the function argument relations. Position in the syntax tree is irrelevant. So, let the conversion map produce a set of operators of the form

(192) (FOR QUANT VAR TYPE/FUNCTION ARGS RESTRICTION)

The VAR, TYPE/FUNCTION, and ARGS are as in the typed predicate calculus. QUANT is a subset of {DEFINITE DISTRIBUTIVE}. Marking specificity is unnecessary. Specificity controls whether or not an np can take a Skolem modifier. After such modifiers are placed, specificity is irrelevant. RESTRICTION is a lambda expression, but its body is a set of these operator chunks. An example should help clarify this new notation.

(193a) Every boy who is in the frat dated
a girl from a certain sorority.

```
(193b)  {  (FOR {DISTRIBUTIVE} X4 SORORITY () T)
           (FOR {DEFINITE DISTRIBUTIVE} X1 BOY ()
                (LAMBDA (Y)
                  { (FOR {DEFINITE DISTRIBUTIVE} X2 FRAT () T)
                     (FOR {} X3 IN (Y X2) T) }))
           (FOR {} X6 KISS (X1 X5) T)
           (FOR {DISTRIBUTIVE} X5 GIRL (X4 X1) T) }
```

When (a) has a different/per reading, ***a girl*** has a Skolem modifier. This is reflected in the converted logical form by the ARGS of the operator binding X5, ***a girl.***

Note that the two predicates, KISS and IN, have been converted to operators. Although this isn't really necessary, it gives the logical form a useful homogeneity.

This converted logical form is input to the formal semantics algorithm. The algorithm has three steps. The first step is simply to sort the operator sets according the predicate

(194) X<Y if the VAR of X is in the ARGS of Y.

Since the function/argument relation is guaranteed to be a partial order by the Functional Principle, forming a total order is always possible. The partial ordering of (193b) is

```
(195)      X1 < X5 < X6
           X2 < X3
           X4 < X5
```

So, one possible total ordering for it is

```
(196)      (  (FOR {DEFINITE DISTRIBUTIVE} X1 BOY ()
                   (LAMBDA (Y)
                     ( (FOR {DEFINITE DISTRIBUTIVE} X2 FRAT () T)
                        (FOR {} X3 IN (Y X2) T) )))
              (FOR {DISTRIBUTIVE} X4 SORORITY () T)
              (FOR {DISTRIBUTIVE} X5 GIRL (X4 X1) T)
              (FOR {} X6 KISS (X1 X5) T) )
```

The second step converts the order list of operators into proper typed predicate calculus. The operators are nested. The operator collars are removed from the predicates KISS and IN. The symbol SPECIFIC is added to all the QUANTs. Thus, (196) is converted by step two into

```
(197)      (FOR {SPECIFIC DEFINITE DISTRIBUTIVE} X1 BOY ()
                          (LAMBDA (Y)
                                  (FOR {SPEC. DEF. DISTR.} X2 FRAT () T
                                       (IN Y X2)))
                (FOR {SPECIFIC DISTRIBUTIVE} X4 SORORITY () T
                     (FOR {SPECIFIC DISTRIBUTIVE} X5 GIRL (X4 X1) T
                          (DATE X1 X5))))
```

The third step is just to evaluate this expression with the formal semantics of the previous section. Note that although both the indefinite operators of ***sorority*** and ***girl*** are inside the distributive operator of ***boy***, the same sorority must satisfy the expression regardless of the boys; but there can be a different girl per boy, because the X1 argument of GIRL makes the constraint of MEMORY ineffective.

The formal semantics of the typed Skolem form depends on two things. First, the functional principle guarantees that the function/argument relation is a partial order, so typed Skolem Form can be converted to typed predicate calculus. Second, typed predicate calculus has an indefinite specific operator, so it can represent same/per readings without nesting the indefinite operator outside the distributive operator.

8.3 IP Form

The previous two sections presented formal semantics for quantification based on FOR loops, a technique pioneered by Woods. Quantification is realized by repeatedly binding a bound variable to an object in the model, and evaluating the rest of the expression. In effect, Woodsian semantics spreads the possible values of the variables out in time. Tarskian semantics, on the other hand, spread the possible values of variables out in space. If a formula has n variables, Tarskian semantics starts with a set of all possible n-tuples of objects. This set is passed up from the leaves of the expression, undergoing intersection, complementation, restriction, or projection depending on the nature of the dominating node -- ie. conjunction, negation, predication, or existential quantification respectively.

The formal semantics of IP form is a combination of Woodsian and Tarskian semantics. The values of IPs are spread out in time, in exactly the same way that Woods formalized the meaning of the universal quantifier. The extensions of indefinite nps, however, are spread out in space, using the possibility set technique that Tarski employed.

The reason existentials are not realized as Foreach loops is inherent in the structure of IP form. There is no existential operator that dominates the predicates that use its bound variable, as in predicate calculus. Nor is there any explicit linkage, as there is in Skolem form, that would allow one to rearrange the logical form into such a structure. Instead, existentials are at the leaves of the trees. Their effects must come from the bottom up, as in Tarskian semantics.

It is no longer possible to use MEMORY to implement specific operators. Since operators are dominated by the predicates which use their values, they never find out which values satisfied the predicates. So there is no way for them to know to return only the satisfactory values the next time they are called.

A new technique will be used to implement specific operators. When a specific operator returns a value, it adds a note saying what its value is. These "assumptions" are passed up through the functions and predicates -- the values of functions and predicates inherit the assumptions of their arguments. IP nodes check that the values returned by one execution of the body of the iteration has the same assumptions as the values returned by previous executions. Values with different assumptions are thrown away. Thus, if a specific operator is beneath an IP, the IP is true only if that specific operator can have the same value regardless of the value of the IP's bound variable.

With this overview, the actual programs that implement the formal semantics will be presented. It should be pointed out the code is missing many important details, eg. how traces are bound. However, the algorithm is so complicated that its comprehension demands suppression of as much detail as possible. The data structures will be presented first, then the algorithm itself.

The input is a simplification of the IP form. Once again, a "smart" conversion rule is required to separate arguments of type functions from class restrictions. An example of the simplified IP form appears in figure 22. As in the typed Skolem form, every node is assigned a variable, even the main predicate KISS. This uniformity makes the algorithm simpler. Also, QUANT is a subset of {DEFINITE SPECIFIC}. The collective/distributive distinction is represented by whether or not the np has been

---

Fig. 22. Simplified IP Form

Each girl kissed a boy.

```
(IP1
   ARGS ((NP1
          VAR X1
          QUANT {SPECIFIC DEFINITE}
          TYPE/FUNCTION GIRL
          ARGS ()
          RESTRICTION T))
   BODY (VP1
          VAR X2
          QUANT {}
          TYPE/FUNCTION KISS
          ARGS ((TRACE X1)
                (NP2
                   VAR X3
                   QUANT {}
                   TYPE/FUNCTION BOY
                   ARGS ()
                   RESTRICTION T))
          RESTRICTION T))
```

raised to be an argument of an IP.

The logical form will be accessed via the functions

(198)	(VAR <node>)	==> a symbol
	(QUANT <node>)	==> subset of {DEFINITE SPECIFIC}
	(TYPE/FUNCTION <node>)	==> a function symbol
	(ARGS <node>)	==> a list of nodes
	(RESTRICTION <node>)	==> a lambda expression
	(BODY <IP node>)	==> a node

The output data structure is called an "extension". An extension is a set of possible values for a node. Moreover, each value has some assumptions attached to it. The assumptions say what values certain variables are assumed to have. More formally, an extension is

(199)	extension	::= { ext/elt ext/elt ... ext/elt }
	ext/elt	::= (value assuming)
	assuming	::= { assumption assumption ... assumption }
	assumption	::= (variable value)
	variable	::= a symbol
	value	::= a set of data base objects

The pieces of extensions will be accessed by functions with the names given above. For example, (ASSUMING <ext/elt>) returns the set of assumptions associate with the given possible value.

Extension will be built up with the aid of two creation functions.

(200a) (MAKE/EXT/ELT value=X assuming=Y)

(200b) (MAKE/ASSUMPTION variable=Q value=R)

Execution of (a) returns an extension element whose value is X and whose assumption set is Y. Execution of (b) creates an assumption that Q has the value R.

The top level function is called EXTEND. To find the extension of an expression, one calls EXTEND on the root node. It will return an extension. If the null set is returned, the expression is false in the given data base. The code for EXTEND, and a closely related subroutine, appear in figure 23.

EXTEND evaluates each of the node's arguments. The cross product of the resulting

Fig. 23. EXTEND

```
(Define EXTEND (NODE)
          (Foreach A InTheList (ARGS NODE) Do
               VALUE ← (EXTEND A)
               Put VALUE OnTheEndOfTheList ARG/VALUES)
          ALL/POSSIBLE/ARG/VALUES ← (CrossProduct ARG/VALUES)
          (Foreach AV InTheSet ALL/POSSIBLE/ARG/VALUES do
               EXTENSION ← (Union EXTENSION (APPLY/NODE/TO/ARGS NODE AV)))
          (Return EXTENSION))

(Define APPLY/NODE/TO/ARGS (NODE ARGS)
          IF (KIND NODE)=IP
               then (APPLY/IP/TO/ARGS NODE ARGS)
               else (APPLY/FUNCTION/TO/ARGS NODE ARGS))
```

---

possibility sets is taken, and returns all possible combinations of argument values. The function or IP is called with each of these arguments, and the resulting possibility sets are merged into one large possibility set. This set is the extension of the node.

The union operation implements existential quantification. For example, consider

(201) Some men wept.

where ***some men*** is given an indefinite nonspecific interpretation. The root node of the logical form for this sentence is the predicate ***weep***, which has one argument, ***some men***. EXTEND evaluates ***weep***'s argument, and gets back a large possibility set -- say

(202) { ({ALAN CHARLES} {}) ({ALAN DAVID} {}) ({DAVID} {}) ...}

Each value is a set containing some men. There are no assumptions since ***some men*** is taken to be nonspecific. Now WEEP is applied to each of these possible value. Mostly, WEEP will return the null set, since few men weep. But if even one man turns out to have wept, then the union over all the results will be non-empty. Hence, extension of the root node will be non-empty, and the sentence true. It is only when all the men are dry-eyed that the union is empty, and the sentence is false. Thus, the union implements existential quantification.

Extending a function is rather similar to the FOR procedure of the previous section.

The biggest difference is the replacement of MEMORY with mechanisms for handling assumptions. The code appears in figure 24.

Since there is no EXPRESSION as in the FOR procedure, the distinction between range and satisfiers is irrelevant. So the result of EXTEND/AND/FILTER is the set of satisfiers of this node. If the node is definite, this set is the only possible value. If it is indefinite, then any of the subsets of the satisfiers is a possible value. The possible values are sets, because only the IP creates distributive interpretations of nps.

The inheritance of assumptions is implemented by MERGE/ASSUMPTIONS/OF/ARGS. If the node is nonspecific, the union of the assumption sets of the args is attached to each possible value. If the node is specific, then an assumption about its value is added by ADD/ASSUMPTIONS.

Fig. 24. APPLY/FUNCTION/TO/ARGS

```
(Define APPLY/FUNCTION/TO/ARGS (NODE ARGS)
        VALUES ← (EXTEND/AND/FILTER
                    (TYPE/FUNCTION NODE) ARGS (RESTRICTION ARGS))
        ASSUMPTIONS ← (MERGE/ASSUMPTIONS/OF/ARGS ARGS)
        If DEFINITE ∈ (QUANT NODE)
            then VALUES ← (MakeSingletonSet VALUES)
            else VALUES ← (Powerset VALUES)
        If SPECIFIC ∈ (QUANT NODE)
            then EXTENSION ← (ADD/ASSUMPTIONS VALUES ASSUMPTIONS (VAR NODE))
            else (Foreach X InTheSet VALUES Do
                          Put (MAKE/EXT/ELT value=VALUES assuming=ASSUMPTIONS)
                              IntoTheSet EXTENSION)
        (Return EXTENSION))

(Define MERGE/ASSUMPTIONS/OF/ARGS (ARGS)
        (Foreach A InTheList ARGS Do
            TOTAL ← (Union TOTAL (ASSUMING A)))
        (Return TOTAL))

(Define ADD/ASSUMPTIONS (VALUES ASSUMPTIONS VAR)
        (Foreach V InTheSet VALUES Do
            Put (MAKE/EXT/ELT
                  value=V
                  assuming=(ConsElement (MAKE/ASSUMPTION variable=VAR value=V)
                                        ASSUMPTIONS))
            IntoTheSet EXTENSION)
        (Return EXTENSION))
```

The action of APPLY/FUNCTION/TO/ARGS will be illustrated by extending *some men at the funeral*. The np has the type function MEN, with one argument, *the funeral*, and no restrictions. EXTEND has already evaluate the node's argument, so ARGS is a list of one extension element. Moreover, *the funeral* is definite and specific, so its extension is a singleton set with one assumption. That is, ARGS contains just the extension element

(203) ({FUNERAL23} { (X2 {FUNERAL23}) })

where X2 is the variable for *the funeral*.

EXTEND/AND/FILTER takes this extension element and applies MEN to it. The result is the set of men who are at FUNERAL23, say {BRAD DAVID}. MERGE/ASSUMPTIONS/OF/ARGS strips off the assumptions and returns them, namely the singleton set {(X2 {FUNERAL23})}.

Since *some men* is indefinite, the possible values are all possible subsets of {BRAD DAVID}, namely {DAVID BRAD}, {DAVID} and {BRAD}. Since the node is nonspecific, no extra assumptions are made. The resulting extension is

(204) { ({DAVID BRAD} { (X2 {FUNERAL23}) })
({DAVID} { (X2 {FUNERAL23}) })
({BRAD} { (X2 {FUNERAL23}) }) }

If *some men* were taken to be specific, the appropriate assumptions would have been added to each possible value, resulting in the extension

(205) { ({DAVID BRAD} {(X1 {DAVID BRAD}) (X2 {FUNERAL23})})
({DAVID} {(X1 {DAVID}) (X2 {FUNERAL23})})
({BRAD} {(X1 {BRAD}) (X2 {FUNERAL23})}) }

where X1 is the variable of *some men*. The only place assumptions are used is in the evaluation of IP nodes.

The procedure for extending IP nodes is, unsurprisingly, a large loop. The code appears in figure 25. Like the FOR program, APPLY/IP/TO/ARGS walks down the set of values, binding a variable to an element of the set, and evaluating the body. There are two major differences, however. One is that IP's can walk down several value sets in parallel. This is to accommodate sentence such as

Fig. 25. APPLY/IP/TO/ARGS

```
(Define APPLY/IP/TO/ARGS (NODE ARGS)
  LOOP: .  BODY ← (BIND/ARGS/TO/FIRST/VALUE/ELEMENT
                         (ARGS NODE) (BODY NODE) ARGS)
      VALUES ← (EXTEND BODY)
      EXTENSION ← (INTERSECT/ALONG/ASSUMPTIONS EXTENSION VALUES)
      ARGS ← (MOVE/TO/NEXT/VALUE/ELEMENT ARGS)
      If ARGS=FINISHED
          then (Return EXTENSION)
          else (Goto LOOP))

(Define INTERSECT/ALONG/ASSUMPTIONS (EXTENSION1 EXTENSION2)
      (Foreach EXT/ELT1 InTheSet EXTENSION1 Do
          (Foreach EXT/ELT2 InTheSet EXTENSION2 Do
                   If (ASSUMING EXT/ELT1)=(ASSUMING EXT/ELT2) then
                      Put (MAKE/EXT/ELT
                             value=(Union (VALUE EXT/EXT1) (VALUE EXT/ELT2))
                             assuming=(ASSUMING EXT/ELT1))
                    IntoTheSet INTERSECTION))
      (Return INTERSECTION))
```

---

(206) Each cork is fastened to each champagne bottle with a prefabricated wire basket.

In this sentence, which was discussed in section 5.2, the IP has two arguments: corks and bottles. To handle multiple arguments, two subroutines are used. BIND/ARGS/TO/FIRST/ELEMENT takes the first element of each value, and binds the appropriate variable to these elements.[1] However, it actually has to build a new extension, including a copy of the assumptions. All this bookkeeping is quite messy, so the subroutine is not presented here. Its operation is hopefully clear: it binds the appropriate trace in BODY to the first element of the appropriate value sets.

Another messy subroutine, MOVE/TO/NEXT/VALUE/ELEMENT, advances the iteration by lopping off the first elements of each of the value sets. If there aren't any more elements, it returns FINISHED, signally that the iteration is complete.

APPLY/IP/TO/ARGS differs from FOR in a second, more substantial way. It is very careful that the assumptions made during one evaluation of the BODY are the same

1. For simplicity, I am assuming that when there are multiple arguments to an IP, the value sets have been ordered by God to correctly correspond to each other. That is, the sets of corks and bottles have been ordered so that the right cork is paired with the right bottle.

as the assumptions made during previous executions. This is the responsibility of the subroutine INTERSECT/ALONG/ASSUMPTIONS. Its action is best illustrated with an example used in the previous section.

(207a) Each girl kissed a boy.

(207b) (KISS MARLA DON)
(KISS LUCY CHARLES)
GIRLS = {MARLA LUCY}
BOYS = {DON CHARLES}

The root node of the logical form for this sentence is an IP node. Evaluating its argument, ***each girl***, returns the extension

(208) { ({MARLA LUCY} {(X1 {MARLA LUCY})}) }

The IP binds X1 to the first element of the value, namely MARLA, and extends ***X1 kissed a boy***. The arguments of KISS are extended next. X1 has only one possible value, MARLA. ***A boy*** has two possible values, DON and CHARLES. So KISS has two possible argument lists, namely (MARLA DON) and (MARLA CHARLES). Application of KISS to the first argument list returns the extension

(209) { ({T} { (X1 {MARLA LUCY}) }

There is only one possible value, T, and no new assumptions have been added. Application of KISS to the second argument list returns the empty set, since Marla didn't kiss Charles. The union at EXTEND merges these possibility sets. The resulting extension, namely (209), is returned to the IP as the value of the BODY when X1=MARLA. Similarly, (209) is returned as the value of the BODY when X1=LUCY, on the next iteration.

It is crucial that ***a boy*** is nonspecific, and therefore added no new assumptions to the extension of KISS. Consequently, intersecting the two extensions of KISS with respect to their assumptions just results in (209) again. Hence, the iteration finishes with a non-empty extension, and the sentence is true.

If the sentence is

(210) Each girl kissed a certain boy.

then the second argument of KISS is specific. Hence, the extension of ***X1 kissed a certain boy*** with X1=MARLA is

(211) { ({T} { (X2 {DON}) (X1 {MARLA LUCY}) }) }

Here, the assumption that X2=DON has been added, where X2 is the variable for ***a certain boy***. Similarly, the extension when X1=LUCY is

(212) { ({T} { (X2 {CHARLES}) (X1 {MARLA LUCY}) }) }

Now X2 is assumed to be CHARLES. When INTERSECT/ALONG/ASSUMPTIONS tries to merge these two extensions, it finds they have different assumption sets. So the result of the intersection is the empty set. Consequently, the IP return the empty set, and the sentence is false.

In short, the assumptions implement the specific/nonspecific distinction. The loop in APPLY/IP/TO/ARGS implements universal quantification. The union in EXTEND and the Powerset in APPLY/FUNCTION/TO/ARGS implement existential quantification.

8.4 Summary

The formal semantics for nonstandard operators of typed predicate calculus were implemented with two simple changes to Woods' FOR function. The Powerset function was used to implement collective operators. Specific operators were given memory, so that they only tried values that satisfied their body the last time they were called.

The formal semantics for typed Skolem form was quite simple. Since the function/argument relation is guaranteed to be a partial order, one simply sorts the nps into a total order. The ordered nps are nested to form an expression in the typed predicate calculus. The operators are all specific, since the extra arguments of Skolem form disable the memory in just the right way.

The formal semantics for IP form was a combination of Tarskian and Woodsian semantics. IPs were implemented as multiple-argument loops. The indefinite nps were implemented by returning possibility sets as extensions. The specific interpretation was implemented by assumptions about the value of the specific nps. These assumptions were passed up, and prevented specific nps from having different values in different evaluations of the bodies of IPs.

All three formal semantics relied on a non-trivial conversion rule. The rule decides whether to translate an np modifier into an argument of the np or into a class restriction. This rule disambiguates quantifier scope when the np is distributive and the modifier contains an indefinite np, eg.

(213a) Every man with a layered haircut is important.
(213b) Every step of a layered haircut is important.

To give (a) a different/per reading, the pp must be converted into a class restriction. If *a layered haircut* is an argument of the np, as in (b), then the sentence has a same per reading in all three logical forms. This is undoubtedly the weakest part of all three theories of quantifier scope.