A Graph-Theoretic Approach to the Semantics of Discourse and Anaphora

Clive John Cox

University of Sussex

D.Phil. Thesis

September 1995

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other University for a degree.

Clive John Cox

Acknowledgements

I am deeply indebted to my two supervisors, Bill Keller and Roger Evans, who have always supplied useful and intelligent comments on the many drafts of this thesis, as well as providing welcome encouragement throughout the research process. I must thank the University of Sussex for financially supporting me with a research bursary.

I would like to thank those people in the School of Cognitive and Computing Sciences who have helped make my time there more enjoyable, especially Jo Brook, Stephen Eglen, Richard Harry and Philip Jones. Outside the university, I would especially like to thank Fernando Salmón whose friendship has helped me most through my years of research and Montserrat Cabré whose support and encouragement have been constant.

Finally, I would like to thank my parents, Mary and Stan Cox, who have always supported me through my years in education.

Abstract

This thesis is concerned with the formal semantic analysis of discourse and anaphora. A formal model-theoretic semantic framework, Graph-Theoretic Semantics (GTS), is developed covering simple extensional English discourse involving singular and plural noun phrase anaphora.

In opposition to previous theories of discourse anaphora, (such as Kamp's DRT or Groenendijk and Stokhof's DPL) anaphoric antecedent information is stored and manipulated within the denotational space. Graph-theoretic denotations are utilized for this purpose. Graph vertices describe individuals identified by the interpretation of a discourse. Graph edges describe constraints amongst the individuals described by the graph vertices. The GTS framework treats these *denotation* graphs as constraint networks in order to correctly handle the various anaphor-antecedent relations that have been proposed within the literature.

The framework is best viewed as a theory of anaphoric analysis. That is, the framework determines how, in certain discourses, appropriate information for anaphoric reference can be derived and, given particular references, how appropriate interpretations can be provided. The framework does not impose a particular set of constraints on anaphoric reference. However, it does provide through the representational and denotational domains the means for providing appropriate constraints on anaphoric reference to allow for the development of particular theories of anaphoric reference. Some example theories of anaphoric reference are provided.

As well as providing a formally precise semantic framework, attention has been paid to theoretical and practical computational issues. The semantic representation is described with unification feature-structures, providing a flexible, powerful and extensible representational foundation for the semantic interpretation. The construction of appropriate semantic representations from an example grammar is illustrated within the PATR unification grammar formalism. The denotational description of language as graphs which are treated as constraint networks allows the extensive research into the efficient solution of constraint satisfaction problems to be utilized. An implementation of the framework is provided.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis concerns the computational semantic analysis of discourses that contain pronominal anaphors which reference information derived from nominal antecedents. Some simple examples are given below.

(1) John owns a Ferrari. He is rich.

(2) Every farmer owns two donkeys. They beat them.

In (1), the **referential** pronominal anaphor *he* has as its preferred **antecedent** the person described by the proper name *John*. In the second discourse, the **bound** pronominal anaphors *they* and *them* have as respective preferred antecedents the farmers and donkeys identified by the first sentence. One plausible reading for the second sentence of this discourse can be paraphrased as stating that *every farmer beats the two donkeys he owns*. The problem of determining the available readings a sentence with anaphoric constituents has is made more complex by the different possibilities available for analysing sub-constituents of a sentence, e.g., the available verbal readings such as distributive, collective (see van der Does (1991)) and the analysis of quantifiers along with their associated scope, (see Barwise and Cooper (1981)). Each different possible interpretational decision will derive information which needs to be collected and be made available for later reference by anaphors. Some more complex examples are shown below.

(3) Some farmers who own a donkey beat it. This would not happen if they were inspected by vets.

(4) Few farmers do not beat a donkey they own.

For (3), there are at least two readings of the second sentence. Firstly, that all the donkeys that are owned and beaten by a farmer should be inspected by a vet. In some sense this information can be derived from the analysis of the entire first sentence. However, there is also a reading in which the second sentence can be said to require that *all donkeys owned by a farmer should be inspected by a vet* not just the ones that are owned and beaten. This reading requires information derived from some sub-sentential constituent of the first sentence, in particular the phrase *farmers who own a donkey*. This is problematic as many theories only keep anaphoric information derived from the analysis of complete sentences. That is, they can only handle the first of these two readings. Meanwhile, the discourse in (4) illustrates the complex interaction of negatives with pronominal anaphors. One appropriate interpretation for (4) can be paraphrased as *most farmers beat the donkeys they own*. An anaphoric semantics must handle the interaction between negatives and anaphoric information. A central purpose of this thesis is to develop a semantics of discourse anaphora which can correctly handle the varied problems found in the analysis of anaphoric discourse.

The thesis can be seen as contributing to the theoretical debate concerning discourse anaphora whose chronological history is given briefly below. In the early 1980's the appearance of Hans Kamp's Discourse Representation Theory (DRT) (1981) and Irene Heim's File Change Semantics (1982) provoked considerable interest by proposing that radical departures were required from previous formal semantic theories, in particular that of Montague semantics (Montague, 1974a; Montague, 1974b), to allow for the proper analysis of discourse anaphora. Work since then has has looked at challenging the radical methodological assumptions of these earlier works, for example, Groenendijk and Stokhof (1990a; 1990b), and Zeevat (1989), and the empirical adequacy of these theories, for example, Roberts (1989), Chierchia (1991), Heim (1990), and Kadamon (1990). In parallel with work on the formal semantics of anaphora there has been increased interest in the formal semantics of plurals, initially encouraged by the work of Scha (1981). Further work has looked at the possible formal theoretical constructs required for the semantics of plurals and extending the empirical predictions, for example, Link (1983; 1984; 1987), Landman (1989a; 1989b), and van der Does (1991). Recently, the two areas have converged with the investigation of discourse anaphora involving plural noun phrases, for example van den Berg (1990), Kamp and Reyle (1993), and Elworthy (1993). The present work contributes to this research, looking at discourse anaphora involving both singular and plural noun phrases.

The thesis also extends the avenue of semantic investigation looked at by Haddock (1992) who utilizes constraint networks for the purposes of semantic evaluation. He derives constraint satisfaction problems (CSPs) from simple existential singular non-anaphoric sentences of English. Techniques from constraint satisfaction can then be utilized to help semantically evaluate these sentences in an efficient manner. However, Haddock notes that[1]:

> The CSP paradigm does not straightforwardly admit quantified sentences such as *Each man loved at least two women* or a treatment of plural reference. .... Examples such as these show that further work is required in the borderland between natural language semantics and constraint network theory.

This thesis contributes to this "borderland" by providing a semantics of natural language in which graph-theoretic denotations are derived for natural language expressions, the semantic interpretation utilizing these graphs as constraint networks. The proposed semantics will cover the two areas highlighted above by Haddock as posing problems for a CSP-based approach, i.e., quantified expressions and plural reference.

The semantics proposed, Graph-Theoretic Semantics (GTS), will also address certain problems arising from the previous treatments proposed by the theories identified above. Of particular concern are the limited range of anaphoric and non-anaphoric readings that are derivable by these theories and the lack of flexibility in the choice of a particular reading from the derivable readings in each discourse situation. Furthermore, most theories have very rigid mechanisms to constrain anaphora making any alternative viewpoint on the constraining of anaphora difficult to impose.

[1] Haddock (1992, p. 419).

There are certain broad methodological and computational concerns that have played an important role in the development of the new semantics and which mark it apart from other theories. These concerns will be recurrent themes throughout the thesis but due to their central role in the critical analysis of previous theories and the formulation of a new semantics they are discussed briefly below.

- Compositionality.
  Compositionality as a methodological principle has been greatly valued by most semantic theorists. Informally, it requires that the meaning of any (semantic) structure should be solely determined from the meaning of the parts of that structure. The advantage of such a methodology is that it induces a modular structure to the semantics with all the consequent advantages of extendibility and modification such a structure entails. Furthermore, such a structure facilitates a clean interface between syntax, semantic representation and semantic denotation. However, as will be discussed, deriving a compositional semantics for discourse anaphora has proved a difficult task.

- Availability of anaphoric information.
  An important aspect of any anaphoric semantics is the construction and storage of appropriate anaphoric information structures. The analysis of a pronoun requires that an appropriate antecedent be found from the anaphoric information structures derived by the semantics from the previous discourse. What anaphoric information is made available is critically dependent on what form of semantic analysis is provided for the non-anaphoric elements of a discourse. Anaphoric information will be affected by the types of reading given to verbs (e.g., distributive or collective), the possible readings and scope assignments given to noun phrases, the various readings given to negation, amongst a variety of other interpretational decisions. How each sentence in a discourse is interpreted with respect to these (and other) degrees of freedom will affect what anaphoric information is derived. Finally, anaphoric information need not only be kept from the analysis of complete sentences but also from the analysis of sub-sentential constituents which assuming a compositional analysis will derive their own semantic denotations which can be utilized for anaphoric purposes, as was shown by the discourse in (3).

- Flexibility.
  As will be discussed in the next chapter there are several proposals concerning the possible relationship between anaphor and antecedent, i.e., several so-called anaphor-antecedent relations. In chapter 3, it will be shown that most theories choose some subset of these anaphor-antecedent relations to implement. Furthermore, they choose some subset (usually quite small) of the available options for analysing non-anaphoric discourse. Restricting a semantics to a small subset of the possible data is certainly good practice when developing a new semantics. However, admirable as this is, it must always be asked how easily such a theory can be extended, if needed, to cover a larger subset of the data and secondly, how easily such a theory can be modified to handle a *different* subset of the data. These concerns are important if one subscribes to the views of Kaplan (1987) who described one of the "seductions" of computational psycholinguistics as follows, (Kaplan, 1987, p.157):

> The substance seduction is the mistaken belief that you know what you're talking about....[T]he problem is that, at least in the current state of the art, they [linguists] don't know which generalizations and restrictions are really going to be true and correct, and which are either accidental, uninteresting or false. The data just isn't in...

If one believes that indeed, "the data just isn't in" then a semantics which perfectly covers a certain subset of data but which is impossibly difficult to modify would seem to lose some of its charm.

- Constraint mechanisms.
  In most discourses, every pronoun can not plausibly refer to every derived anaphoric information structure. Given this, an anaphoric semantics must contain constraint mechanisms to limit the available antecedents for any particular anaphor. Constraining is implicitly related with the previous two concerns. A semantic theory which is as flexible as one would wish and which derives all manner of anaphoric antecedents but which has no means of constraining these possibilities would seem not to be what one would want. Furthermore, the more inflexible and limited the semantics the less powerful its constraint mechanisms need to be as the space of possible anaphoric information structures is more impoverished. Given that at present no perfect constraint mechanism for anaphoric reference is known, a semantics which allows different constraint mechanisms to be incorporated would seem to be preferable over the imposition of any particular one. Therefore, a theory must illustrate that varied constraint mechanisms can be derived and incorporated into the workings of the theory at different possible stages, e.g., during the derivation of acceptable semantic representations or during the semantic interpretation.

- Computational usefulness.
  The research presented within this thesis has been carried out within the field of computational linguistics and, therefore, the computational properties and computational applicability of any linguistic framework are important. There are several different metrics that might be looked at to ascertain a framework's computational usefulness, for example, the computational complexity of any resulting framework, the ease of which a framework can be implemented and the ease of which it can be integrated with existing computational mechanisms and theories.

Driven by these methodological issues the proposed new semantics, Graph Theoretic Semantics, will be a framework for constructing empirical theories of anaphora, rather than an empirical theory itself. That is, the theory will provide the means to construct and manipulate the information structures required for anaphoric analysis but will not provide restrictions on anaphoric reference itself. However, the framework will provide an appropriate set of "parameters" which will allow the construction of empirical theories of anaphora. A clear distinction should thus be made between:

1. A framework for the construction of empirical theories of anaphora

2. An empirical theory of anaphora

GTS is the first of these two possibilities. A related distinction used throughout this thesis is that of anaphoric analysis and anaphoric reference. **Anaphoric reference** is the particular set of relations allowed between anaphors and antecedents in discourse. While, *anaphoric analysis* will be viewed as the methods and means by which information structures, (in this case model theoretic information structures), are derived and manipulated for the purposes of allowing particular anaphoric references.

Chapter 2 will look at the theoretical background of discourse anaphora and its semantic analysis, while chapter 3 will look in detail at particular semantic theories of discourse anaphora. In chapter 4, a new theory of discourse anaphora will be introduced. Chapters 5 and 6 will discuss the theory in detail, giving in chapter 5 a non-anaphoric semantic theory of discourse which in chapter 6 is extended to handle anaphoric discourse. Chapter 6 will also contain a fully worked example of the derived semantic framework, in section 6.3. This worked example may be useful to the reader to better understand the workings of each part of the semantic framework as it is introduced. Chapter 7 will look at implementational issues for the new theory.

Chapter 2

Anaphora and Semantics

This thesis is concerned with providing a formal semantic account of one aspect of discourse anaphora known as pronominal noun phrase anaphora. The language for this study will be English and I shall only be looking at extensional discourse, i.e., discourse without explicit or implicit modal or belief operators. The research work has been conducted within the domain of computational linguistics and therefore an ancillary aim has been to provide a formal framework that can be accommodated within and used by today's body of computational theories. This chapter, however, is dedicated to providing a review of the linguistic analysis of noun phrase anaphora and the contributing linguistic phenomena that play an important role within the thesis.

To begin with, definitions must be provided for the basic terms of *discourse* and *anaphora*. For the purposes of this work I will define a discourse as simply a temporally ordered set of sentences. Hirst (1981) in his overview of anaphora understanding defines anaphora as:

> The device of making in discourse an abbreviated reference to some entity (or entities) in the expectation that the perceiver of the discourse will be able to disabbreviate the reference and thereby determine the identity of the entity.

The entity referred to may have been evoked implicitly or explicitly by the preceding discourse. This definition covers a wide range of linguistic constructions. However, I will only be concerned with those instances of anaphora where the anaphoric device is a third person pronoun and the entity referred to is derived from the semantic analysis of one or more nouns within the discourse. The pronoun creating the anaphoric device within a discourse I shall call the *anaphor*. The entity or entities referenced I shall call the *antecedent*. This use of the word antecedent should not be confused with its use to refer to the actual word or words within the discourse for which the entities to which the anaphor refers seem to be derived[1]. I will always use the former definition for the term antecedent while using the term *antecedent phrase* to refer to the actual word or words (usually a noun phrase) from which the entities identified by the antecedent are derived. Hirst's definition needs to be further constrained for the present purposes as I shall in the main

[1] This secondary use is sometimes compounded within the literature by the use of identical subscripts within syntactic analyses of sentences containing anaphoric references to mark the words which seem to provide an anaphoric link, e.g. Every soprano$_1$ loves herself$_1$. The unfortunate aspect of co-indexing is that it forces the viewpoint that the antecedent phrase alone provides the informational content for the antecedent instead of seeing the antecedent as derived from the informational content of the antecedent phrase as it *interacts* with the rest of the information within the surrounding sentence through an interpretation.

only be discussing cases of anaphora in which the anaphor occurs temporally after the antecedent within a discourse. The related term cataphora will be used when discussing cases in which the anaphor occurs temporally before the antecedent. Furthermore, I shall presume all antecedents to be explicitly derivable from the discourse and I will therefore not be discussing occurrences of exophora in which the antecedent is derived in part or in whole from information external to the written or spoken discourse.

There are a wide range of possible links that can exist between an anaphor and an antecedent but researchers have identified certain types of *anaphor-antecedent relation* as basic and it is these relations that I shall discuss in the next section.

2.1 Basic Anaphor-Antecedent Relations

Linguists have proposed several types of anaphor-antecedent relation resulting in the classification of several different types of pronoun. However, it is of critical importance to remember that these pronominal types simply reflect different underlying anaphor-antecedent relations. It is unfortunate that the standard linguistic terminology invariably speaks about pronominal types instead of anaphor-antecedent relations, thereby shifting the focus away from the complexity of anaphoric relations and onto only one structure of the overall phenomenon, i.e., the pronoun. This unintentional focusing I will show has been reflected in disjointed analyses of anaphor-antecedent relations based primarily on aspects of either the anaphor or the antecedent but not where the true analysis lies within the whole anaphoric relation.

The discourses below show examples of the two most basic types of pronoun. Example (5) contains a **referential** pronoun *he*, while (6) contains a **bound** pronoun *she*.

(5) John owns a Ferrari. He likes driving.

(6) Every soprano thinks she sings beautifully.

Referential pronouns refer to a unique identifiable entity or collection of entities from the discourse. In (5), *he* refers to the unique entity identified by the name *John*. Bound pronouns meanwhile are so named because they seem to act in a similar way to bound variables in logic and range over a set of entities. In (6), *she* refers to each individual soprano identified and "bound" by the quantifier expression *every soprano*.

There has long been a debate about whether this division accurately reflects all the data. One prominent dissenting voice is that of Gareth Evans (Evans, 1977; Evans, 1980; Evans, 1985) who proposes that referential pronouns should be split between those as found in (5) which are **coreferential** with a previous definite description and a new type of referential pronoun known as an E-Type pronoun whose antecedent is derived from a quantified expression. Evans' complete taxonomy is shown below.

1. Deictic (exophoric) pronouns.
2. Coreferential pronouns.
3. Bound Pronouns.
4. E-type pronouns.

Evans provides data which he thinks shows that E-type pronouns are a form of anaphor-antecedent relation occurring within English sentences which resemble aspects of both referential and bound pronouns but which can not be said to be either of these forms. Within his 1980 paper he provides the following sentences as examples which require an E-type analysis:

(7) John owns some sheep and Harry vaccinates them in the Spring. (p. 339)

(8) Every villager owns some donkeys and feeds them at night. (p. 353)

Evans argues that the pronoun *them* in (7) can not be viewed as a bound pronoun as this would require that the sentence had a meaning equivalent (by his understanding of bound pronouns) to the paraphrase below, a reading which he believes is unavailable.

(9) Some particular group of sheep are owned by John and are vaccinated by Harry in the Spring.

Evans believes this is an incorrect reading as the sheep identified in (9) may only be a strict subset of all the sheep owned by John, rather than all John's sheep, which is Evans preferred reading of (7). Evans sees these new pronominal types as rigid designators which have their reference fixed by a definite description recoverable from the antecedent. Evans argues against making his pronouns 'go proxy' for the recoverable description as this would then allow them to be ambiguous in certain sentences such as (10).

(10) A man murdered Smith, but John does not believe that he murdered Smith.

As (10) contains a propositional attitude an analysis which allowed the anaphor *he* to go proxy for a definite description derived from its antecedent would allow both *de re* and *de dicto* readings. The *de re* reading would attribute to John the belief that the man identified as the murderer is not the murderer. While the *de dicto* reading would attribute John to belief that the man who murdered Smith did not murder Smith. As the *de dicto* reading seems self-contradictory, Evans assumes it shows that E-type pronouns do not go proxy for a description but have their reference fixed by a description derived from the antecedent. However, Neale (1990) investigates changing the E-type analysis to one in which these pronouns actually do go proxy for the definite description derived from the antecedent. These pronouns he calls D-type pronouns. Against sentences like (10), Neale provides sentences such as (11).

(11) A man murdered Smith. The police have reason to think that he injured himself in the process. (p. 186)

This example seems to allow both a *de re* and *de dicto* reading. Neale suggests that the *de dicto* reading of (10) is indeed available but is rejected simply on commonsense grounds. As the distinctions between E-type and D-type pronouns generally require one to look at sentence constructions outside the extensional domain I will not be concerned with the different predictions between these proposals and simply assume Evans' E-type anaphor-antecedent relations are the sole standard bearers from within the "pronouns as definite descriptions" camp.

Evans continues his proposal by claiming that bound and co-referential pronouns can be provided with a similar semantics. Indeed he appeals to the reader that:

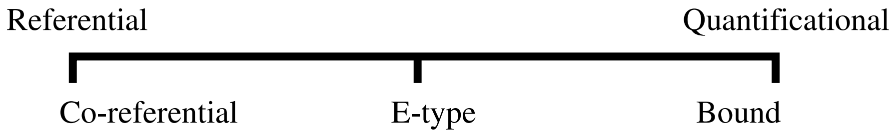


Figure 2.1: Anaphor-Antecedent Relations.

> It is simply not credible that the speaker's capacity to understand the sentences *John loves his mother*, *Harry loves his mother* ...is in no way connected with his understanding of the sentences *No man loves his mother*, *Every man loves his mother*[2].

As the differences within his examples lie most obviously in the antecedent he claims that both type of pronoun can be treated in a coreferential manner by defining how the antecedent phrase satisfies the complex predicate derived from the rest of the sentence in a Fregean truth-conditional manner. We can look at (12) and (13) as complementary examples.

(12) John loves his mother.

(13) Every man loves his mother.

The complex predicate is *x loves x's mother* and we must either bind the referent of *John* to this predicate and check for truth or we must bind each applicable man defined by *every man* to the predicate in turn and check for truth. He attacks Lasnik (1976) who had previously proposed that deictic and referential pronouns should be treated similarly under a pragmatic framework primarily because each type of pronoun refers to some salient entity within the discourse be that textual or through the surrounding context. Evans protests that this conjoining of deictic and referential pronouns would disallow any of the similarities he observes between bound and co-referential pronouns from being treated uniformly. That is, Lasnik's and Evans' work are incompatible[3].

Evans' work has highlighted the inadequacy of certain logical based translations in providing a unified account of anaphoric semantics. E-type pronouns can also be seen as showing that pronominal anaphor-antecedent relations exist which can viewed neither as purely referential or purely quantificational in nature. By this view anaphor-antecedent relations can be seen as ranging along a line as shown in figure 2.1. Most theories begin with the intention of either providing a referential or bound (quantificational) analysis of pronouns. Any semantic theory of anaphora must at least provide solutions to the problems that promoted these various forms of pronouns to be proposed. That is, the different pronominal classifications can be seen as identifying various forms of anaphor-antecedent relation which (English) discourse can derive. It is these anaphor-antecedent relations that a theory must handle.

The next section will look at one class of sentence which Evans would class as containing E-type pronouns which has driven the theoretical debate within recent years with the creation of several new theories attempting to handle their complex semantics.

---

[2]Evans, 1977, p. 345

[3]Evans' proposed reasons why his treatment should be preferred over Lasnik's can be found in Evans (1980).

2.2 Donkey Sentences

A great deal of theoretical discussion has centered around finding the proper analysis of one group of sentence constructions commonly known as *donkey sentences* (Geach, 1962)[4]. The two classic examples are shown in (14) and (15).

(14) Every farmer who owns a donkey beats it.

(15) If a farmer owns a donkey he beats it.

Following Elworthy (1993), I shall call donkey sentences like (14) *quantified* donkey sentences while those donkey sentences like (15) I shall call *conditional* donkey sentences. Heim (1988) provides a more formal definition for these sentence constructions as:

> ...sentences that contain an indefinite NP which is inside an if-clause or relative clause, and a pronoun which is outside that if-clause or relative clause, but is related anaphorically to the indefinite NP. (p. 44)

The problem with these sentences is that it is not clear what anaphor-antecedent relations they contain. Analysing (14), we can be certain that the relationship between the pronoun *it* and the antecedent derived from analysis of the antecedent phrase *a donkey* within the sentences is not coreferential, as the antecedent is derived from within a quantified construction. This leaves us with either choosing a bound or E-type anaphor-antecedent relation. Adding to the problems these sentences provide is the fact that readers do not have consistent intuitions about what they actually mean, especially when read in isolation as above. I shall discuss in the remainder of this section the basics of both the bound and E-type analyses of these sentences and show how the different analyses have promoted different "readings" of these sentences to suit the particular analysis provided.

When theorists attempt to give a bound analysis to a pronoun they must specify how the antecedent binds the pronoun and decide upon a formalism to identify how the pronoun falls within the scope of the antecedent and how each is to be represented. Traditionally, the formal language used has been first-order predicate calculus or extensions of first-order predicate calculus. This close proximity between the discussion of bound anaphor-antecedent relations and the description of those relations within first-order predicate calculus has caused the distinction between the two to be blurred. This has had the effect that when bound pronouns are discussed it is assumed that the nature of the binding effects, the way quantifier scope effects are handled and the contributions both the anaphor and antecedent play within the relation are to be understood with the meanings these terms inherit from our understanding of predicate calculus. However, predicate calculus need not be the only way in which we understand binding and scope effects for bound pronouns.

Given that the traditional analysis of bound pronouns has been illustrated, discussed and analysed using first-order predicate calculus style languages I shall discuss the problems of a bound analysis of donkey sentences using first-order predicate calculus. The traditional stance is that indefinite noun phrases such as *a donkey* should be translated into predicate calculus with an existential quantifier and thus (14) will be translated as (16).

[4]Geach takes these sentences from medieval manuscripts, although similar sentences were discussed by the stoic philosopher Chrysippos.

(16) $\forall x[(farmer(x) \wedge \exists y[donkey(y) \wedge own(x,y)]) \rightarrow beat(x,y)]$

However, the variable y in $beat(x,y)$ occurs outside the scope of the existential quantifier as a free variable. This translation under the standard semantics for predicate calculus will not provide an appropriate meaning for (14). One solution would be to translate the indefinite noun phrase *a donkey* with universal quantification[5] and thus provide a translation as given below.

(17) $\forall x,y[(farmer(x) \wedge donkey(y) \wedge own(x,y)) \rightarrow beat(x,y)]$

Under the standard semantics for predicate calculus (17) provides the so-called **universal** reading for donkey sentences which has a paraphrase for (14) shown in (18):

(18) **Universal Reading** for (14).
Every farmer who owns one or more donkeys beats every donkey he owns.

The fact that one can indeed provide a bound anaphoric reading for (14), a sentence which Evans would classify as containing an E-type pronoun, shows that what constitutes our understanding of bound anaphora clearly depends on the representational language we use to display the "bound" anaphoric effects. That is, the empirical effects of "bound" anaphoric references are dependent on the manner in which a semantic analysis expresses the nature of the binding between anaphor and antecedent.

If we accept the translation in (17), we are left with as a consequence the conclusion that indefinite phrases such as *a donkey* are ambiguous between an existential and universal interpretation within a compositional interpretation. Robin Cooper (1979, p. 81) objects to this proposed ambiguity in the analysis of indefinites stating that we would then have to accept that one interpretation of the sentence in (19) has an equivalent meaning to (20).

(19) A donkey died.

(20) Every donkey died.

The manner in which theories such as Discourse Representation Theory (Kamp, 1981; Kamp & Reyle, 1993) and Dynamic Predicate Logic (Groenendijk & Stokhof, 1991b) circumnavigate this problem will be discussed within the next chapter.

For Evans, the pronoun within a donkey sentence is a classic example of an E-type pronoun and should be interpreted as defining a definite description derivable from the antecedent. The standard assumption following Russell (1905) is that definite descriptions have uniqueness effects. This would therefore force a reading of the donkey sentence in (14) which is paraphrased below. This reading I shall call the **unique antecedent** reading following Elworthy (1993).

(21) **Unique Antecedent** Reading for (14)
Every farmer who owns a single donkey beats the (single) donkey he owns.

One example given by Cooper (1979, p. 81) to back up the unique antecedent reading is provided below.

(22) Every man who has a daughter thinks she is the most beautiful girl in the world.

[5] A solution first proposed by Geach (1962).

Cooper protests that (22) simply doesn't say anything about a man who has more than one daughter and particularly it does not commit such a man to "the contradictory belief that each of his daughters is the most beautiful girl in the world"[6]. Nevertheless, my intuitions suggest that this example has more to do with the definite interpretation of the phrase *the most beautiful girl in the world* which seems to imply that there can only be one such girl. This implication then forces any reader to restrict each man to having only one daughter. Added to this, within a suitable context one can alternatively view the statement *the most beautiful girl in the world* as an affectionate rhetorical device on the part of the speaker. For example, one could imagine a conversation as below.

(23) "John spoils his two daughters so much. He has just bought them both expensive ball gowns! I suppose, every man who has a daughter thinks she is the most beautiful girl in the world."

My intuitions give the donkey sentence within (23) a universal reading.

Cooper applies an E-type analysis within an extended Montagovian framework. This allows him to define the semantics of the pronoun within donkey sentences as a type of contextual function which provides a more flexible interpretation for the definite description attributable to the pronoun. This in particular allows him to provide a reading for the donkey sentence in (14) which simply requires that a unique donkey owned by the farmer is contextually salient, but not that each farmer must own a single donkey. Following Elworthy again, I shall call this reading the **unique anaphor** reading which for (14) has a paraphrase shown below.

(24) **Unique Anaphor** reading for (14)
Every farmer who owns one or more donkeys beats one of the donkeys he owns.

One contextual situation which might highlight more directly a unique anaphor reading is given below.

(25) Every parishioner who has a teapot uses it when the vicar calls for tea.

This example seems to me to allow parishioners to own more than one teapot but to use a particular one when the vicar calls for tea.

Schubert and Pelletier (1989) provide evidence for another reading of donkey sentences within a paper concerned with the wider issue of generics.

(26) Everyone who has a donkey must donate its services for one day during the festival. (p. 200)

(27) Every man who owns a donkey will ride it to town tomorrow. (p. 201)

Schubert and Pelletier do not believe that these sentences can have universal readings, insisting that we can't presume, for (26), that wealthy farmers with two or more donkeys are forced to donate all their donkeys or for (27) that farmers must ride *all* their donkeys to town. They also argue against the unique antecedent reading suggesting that it absurd to think that if (26) was ordered by the local government that farmers with two or more donkeys could plead that they

[6] Cooper (1979, p. 81).

were exempt on that ground alone. Schubert and Pelletier, therefore, propose a new reading which they call the **indefinite lazy** reading which applied to the donkey sentence in (14) would have a paraphrase shown below.

(28) **Indefinite Lazy** reading for (14).
Every farmer who owns one or more donkeys beats one or more donkeys that he owns.

Interestingly, Schubert and Pelletier do not use the standard donkey sentence in (14) as an example to which the indefinite lazy reading applies. In my opinion this reading seems to suit most convincingly an isolated reading of (14). Furthermore, they do not provide a critique of the unique anaphor reading which seems to me to most closely fit my reading of (26) and (27).

As has been noted, Schubert and Pelletier are concerned with investigating generics and as such most of their examples contain modal or tense operators, as can be observed in both (26) and (27). Indeed, they treat the standard conditional donkey sentence of (29) as a generic.

(29) If a farmer owns a donkey he beats it.

The evidence presented by Schubert and Pelletier in their analysis of generics suggests to me that we should be wary of providing unified analyses of donkey sentences which contain modal and tense operators along with those like (14) that are basically extensional in nature. Other theorists (Kratzer, 1979; Roberts, 1987; Roberts, 1989) treat *if-then* constructions as modal propositions, while Heim (1988; 1990) and Elworthy (1993) provide further evidence that conditional donkey sentences provide problems separate from their quantified cousins[7]. However, some theorists (Kamp, 1981; Groenendijk & Stokhof, 1991b) do provide an identical extensional semantics for both forms of donkey sentence. My view is that this latter view is misguided and that an analysis of conditional donkey sentences should be conducted within a semantics that provides analyses of modal and tense operators. As this work is strictly devoted to the analysis of extensional discourses only, conditional donkey sentences will not be discussed further. Therefore, in the following discussion, the phrase *donkey sentence* will be taken to mean a donkey sentence of the extensional quantified variety.

2.2.1 Donkey Sentence Readings and Context

Within the previous section I have discussed proposals providing four different types of reading for the donkey sentence in (14). These proposals are repeated below.

(30) Every farmer who owns a donkey beats it.

1. **Universal Reading** for (30).
   Every farmer who owns one or more donkeys beats every donkey he owns.
2. **Unique Antecedent** Reading for (30)
   Every farmer who owns a single donkey beats the (single) donkey he owns.
3. **Unique Anaphor** reading for (30)
   Every farmer who owns one or more donkeys beats one of the donkeys he owns.

[7] See Elworthy (1993, pp. 19-23) for a summary of different analyses provided for conditional donkey sentences.

4. **Indefinite Lazy** reading for (30).
   Every farmer who owns one or more donkeys beats one or more donkeys that he owns.

The last section has shown that depending on how we classify the pronoun in (30) (i.e., either as bound or E-type) different anaphor-antecedent relations are promoted. A bound analysis readily provides the universal and indefinite lazy reading while an E-type analysis readily provides either the unique antecedent or unique anaphor readings.

One aspect which connects all the readings provided by different theorists is that each one is either put forward as the single correct analysis for all quantified donkey sentences or one particular reading is promoted with the excuse that the other alternatives are less favoured amongst the theorists "informants" or occur within obscure examples and should therefore be seen as maverick readings and relegated to the periphery. The position advocated here, however, is that no one reading correctly covers all the possible contexts and situations in which quantified donkey sentences occur. I believe the readings are sanctioned in different circumstances by the contextual and pragmatic effects surrounding a particular use of a quantified donkey sentence. The difficulty usually arises when donkey sentences are read in isolation thereby forcing the reader to create a context which produces the promotion of one possible reading. It is my suggestion that a semantic theory that hopes to cover extensional donkey sentences should provide **all** possible readings and leave the task of selection to the pragmatic domain.

In further defence of this claim, presented below are four discourses all containing the standard quantified donkey sentence of (30). Each discourse in my view presents a different reading as the *most likely reading* of the donkey sentence due to the influence of the surrounding context.

(31) **Indefinite Lazy Example**
The farmers of Ithaca are stressed out. They are constantly arguing and often even beat each other. To put an end to it, they go to the local psychologist who recommends that rather than beating each other, every farmer who owns a donkey beats it. They follow her advice, and things improve[8].

(32) **Universal Example**
The farmers of Ithaca are a cruel and callous lot. They treat their farm animals with scant respect. For instance, every farmer who owns a donkey beats it.

(33) **Unique Anaphor Example**
The farmers of Ithaca are a superstitious lot. They believe that at the first sunrise of each year the God of farming must be placated. Therefore, as the sun rises they assemble before the temple, each with one of their farm animals. The rules determine what act they must perform. For instance, every farmer who owns a donkey beats it.

(34) **Unique Antecedent Example**
The farmers of Ithaca are allowed to own one pack animal only. But their treatment of these animals seems bizzare considering the scarcity of the resource. For instance, every farmer who owns a donkey beats it.

[8]Adapted from Chierchia (1991, p. 54).

The examples provided for the *unique antecedent* and *unique anaphor* readings are certainly more strained for the traditional quantified donkey sentence. However, even if these examples do not wholly convince, the array of possible discourse involving other "donkey" sentences certainly suggests that a semantic theory providing only a subset of the possible readings will be unable to correctly interpret certain discourses.

In conclusion, I believe there is no good *semantic* reason for limiting an analysis of donkey sentences to only one reading. The readings that are available seem to be motivated by contextual expectations which may be interpreted differently by each individual person.

2.2.2 Donkey Sentence Readings and Anaphoric Relations

Within this section I shall investigate the question of how the different readings provided for donkey sentences relate to the anaphor-antecedent relations discussed within Section 2.1. The previous two sections have shown us that each reading has been provided within proposals which promoted a particular viewpoint on the analysis of anaphor-antecedent relations. The universal reading has been mainly promoted by Kamp (1981; 1993), Heim (1988) and Groenendijk and Stokhof (1990a; 1991b) whose theories provide universal quantification to the indefinite antecedent phrase within a donkey sentence. Both these theories can be viewed as providing bound anaphor-antecedent relations whose semantic interpretation follows the essence of my naive first-order predicate calculus translation of the standard quantified donkey sentence shown previously in (17). Cooper (1979) who promoted the unique antecedent and unique anaphor readings provided a Montagovian semantics based on the theoretical exposition of the E-type pronouns of Evans. Meanwhile, Schubert and Pelletier prefer an indefinite lazy reading of donkey sentences which treats the indefinite noun phrase as an existentially quantified expression and thus the pronouns within donkey sentences also as bound anaphor-antecedent relations. The four readings are realised by semantic theories which provide particular (bound or E-type) anaphor-antecedent relations. It is my opinion, therefore, that we should view the discussion of donkey sentence readings as a discussion really about different ways of fleshing out bound or E-type anaphor-antecedent relations. The problem for a semantics (like the one I shall propose) which wishes to allow all four types of reading/anaphor-antecedent relation is to provide a unified account of both bound and E-type anaphor-antecedent relations.

The fact that the different anaphor-antecedent relations within donkey sentences have been discussed as different readings provides a clue as to where the real location for conducting the analysis of anaphor-antecedent relations should be. If we look at the standard quantified donkey sentence (shown again below) and reconsider the different readings, the important consideration is how we determine the analysis of the verbal relations in this sentence. In particular, how we decide how many donkeys are owned and beaten by the farmers.

(35) Every farmer who owns a donkey beats it

Table 2.1 shows the number of donkeys that can be owned and beaten by a farmer and the resulting donkey sentence reading this implies. Following, Kamp (1991) and Chierchia (1991) I shall use the term **strong** reading to refer to the universal reading and **weak** reading to refer to the Indefinite Lazy reading. To obtain each reading a semantics must *firstly* provide an analysis of verbal relations that can check the required number of donkeys are owned and beaten and *secondly* provide an analysis which when checking if a donkey is beaten by a particular farmer

x Donkeys Owned	*y Donkeys Beaten*	*Reading*
$x = 1$	$y = 1$	Unique Antecedent
$x \geq 1$	$y = 1$	Unique Anaphor
$x \geq 1$	$1 \leq y \leq x$	Weak (Indefinite Lazy)
$x \geq 1$	$y = x$	Strong (Universal)

Table 2.1: The possible constraints on the verbal relations **own** and **beat** for each *farmer* and the resulting donkey sentence reading.

checks that the same donkey is owned by that farmer. The second requirement essentially means that a semantics must maintain somehow the dependencies on different individuals (in this case, donkeys and farmers). The underlying theme of both E-type and bound pronouns is that they depend on previous relational dependencies within the surrounding discourse or sentence. E-type pronouns, however, have an implicit uniqueness assumption derived from their analysis as Russellian definite descriptions. This can be observed within Table 2.1 where the two E-type anaphor-antecedent relation readings (unique anaphor and unique antecedent) require either $x = 1$ or $y = 1$. The analysis to be outlined within this work is that anaphor-antecedent relations are best handled within the semantic analysis of verbal relations. For the standard quantified donkey sentence the important decisions arise when we determine which farmers own and beat their donkeys and this can only be determined when we analyse the verbal relations themselves.

Donkey sentences are interesting because they provide a succinct example of the interaction between verbal readings and anaphor-antecedent relations. We can view the unique antecedent and unique anaphor readings as readings in which the verbal analysis imposes a uniqueness constraint along with either a weak (indefinite lazy) or strong (universal) anaphoric relation. In terms of the standard quantified donkey sentence this means that whether a farmer beats all (strong reading) or just some (weak reading) of his donkeys is immaterial as he is only allowed to own and/or beat a single donkey. Under this viewpoint there are only two types of anaphor-antecedent relation: the **strong** and **weak** variety. The two E-type readings are reconstructed by imposing either anaphor-antecedent relation along with an appropriate uniqueness restriction on the appropriate verbal relation.

In summary, this section has proposed that bound and E-type anaphor-antecedent relations are best handled within the analysis of verbal relations and that we only require the weak and strong anaphor-antecedent relation. The unique antecedent and unique anaphor readings can be derived through applying an appropriate uniqueness restriction to the appropriate verbal relation and imposing either a weak or strong anaphor antecedent relation.

2.2.3 Uniqueness Restrictions within Donkey Sentences

Within this section I will review a position which attacks the flexible anaphoric basis proposed within this work. The debate concerns the requirement for a uniqueness restriction to be imposed on definite descriptions and by inclusion also anaphors. Russell (1905) believed that definite descriptions carried a uniqueness implication and this is what distinguished them from indefinite descriptions. As pronouns can be thought of as definite descriptions we should find uniqueness implications arising within anaphoric discourse. For Evans and Russell the unique anaphoric referent for a pronoun is determined strictly from descriptive information derived from the antecedent. Interesting complications arise when the pronominal antecedent is derived from an indefinite noun phrase. This is what we see with the E-type pronouns of Evans which leads him following the uniqueness position to say that the pronoun in (36) refers to the maximal set of sheep owned by John and the pronoun in (37) to the only doctor in London.

(36) John owns some sheep and Harry vaccinates them.

(37) There is a Doctor in London and he is Welsh.

For my intuitions (37) doesn't imply there is only one doctor in London. However, Kadamon (1990) looks more closely at this strict understanding of uniqueness and although she finds (37) acceptable she believes (38) shows this treatment of uniqueness is too strict.

(38) A wine glass broke last night. It had been very expensive.

The discourse in (38) does not entail only one wine glass broke last night, but (in Kadamon's view) one particular one that was expensive. Kadamon provides another example as below.

(39) Every chess set comes with a spare pawn. It is taped to the top of the box.

Kadamon again believes that the anaphor-antecedent relation singles out some unique pawn. As she explains:

> For example, if we have been talking about special bonuses, it could be the only one that comes as a special bonus (in addition, perhaps, to the usual two spare pawns). The important thing is that it has to be unique in **some** way, and unique **relative to a choice of chess set**. (p. 283)

Kadamon therefore proposes a so-called *realistic* uniqueness restriction for definite phrases in which:

> Implicated, accommodated, and contextually supplied material may play a role in satisfying uniqueness, and hence in determining what maximal collection (= unique set) is referred to. (p. 286)

She is therefore proposing either a unique antecedent or unique anaphor reading for anaphor-antecedent relations within donkey sentences. However, which one she assumes is correct for a particular situation seems arbitrary. For (39) she believes it may be possible for this discourse to be made felicitously even if the chess sets in question have more than one spare pawn. Some contextual property may highlight the unique spare pawn (per chess set) discussed. Thus, (39) is given a pragmatically determined unique anaphor reading. However, for the quantified donkey sentences below Kadamon enforces only a unique antecedent reading.

(40) Every farmer who owns a donkey beats it.

(41) Most women who own a cat talk to it.

In particular, Kadamon believes that (40) says nothing about farmers who own more than one donkey, and (41) says nothing about women who own more than one cat. Discourses in which she determines one reading should be provided instead of another seem to be arbitrary, or at least in a manner which helps her uniqueness proposition.

Passing over this however, Kadamon comes up against a further problem for her theory exemplified by an example attributable to Heim (1988, p. 89) given below.

(42) Everybody who bought a sage plant here bought eight others along with it.

Here it seems illogical to enforce a particular sage plant as the antecedent for the pronoun. Kadamon's response is to loosen the realistic uniqueness restriction still further to allow cases in which the choice of antecedent can not effect the truth conditions. For (42) to be true it doesn't matter which of the nine sage plants is the referent as all provide the same truthful interpretation to the discourse. However, this is very similar to situations in which the strong (universal) anaphor-antecedent relation is satisfied. For example, for the standard quantified donkey sentence (40) if a strong reading is satisfied then each farmer beats all the donkeys he owns and indeed it would not matter which of these donkeys we chose to satisfy the sentence, as the sentence would still be true. Thus, Kadamon has inadvertently allowed the existence of universal readings, for they are simply the readings in which a uniqueness account is satisfied no matter what (applicable) antecedent is chosen as the referent. Of course the advantage of independently stipulating such a reading is that we can distinguish it from the more limited unique anaphor readings.

This simply leaves us then with justifying the validity of the indefinite lazy reading against the uniqueness attack. The indefinite lazy reading seems to work best in situations in which a verbal relation is attributing a general property rather than some individual property. For instance given the two sentences below.

(43) Every farmer who owns a donkey beats it.

(44) Every man who owns a donkey will ride it to town tomorrow.

If we want to identify farmers who beat donkeys (in (43)) or men who ride to town on a donkey (in (44)) it would seem ridiculous to reject the sentences because some farmers beat several (but not all) of their donkeys and some men ride several (but not all) of their donkeys to town. It is in these situations where we wish a looser reading of a discourse that an indefinite lazy reading becomes appropriate. However, in a situation where we must analyse any prospective satisfying situation in a fine-grained individualised manner the indefinite lazy reading seems inappropriate. For instance given (45) and a situation in which some farmers only brand a few of their donkey an indefinite lazy reading would satisfy this situation although to me a universal reading seems more appropriate here.

(45) Every farmer who owns a donkey brands it with his personal symbol.

Indefinite lazy readings also seem (as has been mentioned before) to work well in discourses with temporal or propositional attitude expressions.

Within this section I have shown that the uniqueness implication for definite descriptions (and pronouns in particular) is either blatantly false or at best needs to be phrased in such a loose and vague manner as to be of little or no semantic interest. Any uniqueness effects that do exist are explainable only with a deep understanding of contextual and pragmatic influences. Any uniqueness effects can certainly not be enforced across the board in an indiscriminate manner.

2.3 Plural Anaphora and Verbal Readings

The framework constructed within this work will be dealing with singular and plural anaphors as well as antecedents derived from noun phrases covering the full range of unary generalised quantifiers. Within this section I shall discuss the different proposals that have been given for the formulation of a model-theoretic semantics of plurals and in particular the different verbal readings that have been provided for verbal relations whose arguments include plural noun phrases. This discussion, although at first sight complementary to a semantics of anaphora, can not be disregarded as the analysis of plural constructions including the possible readings of sentences containing plurals will greatly determine the antecedents available from a discourse and the dependencies which are imposed on those antecedents.

The discussion will be centered around the general linguistic tool of model-theoretic semantic analysis, a style of semantic analysis that will be followed within this work. Central to such a study is the definition of a model structure that provides a description of the world. A standard model-structure is a pair $\langle D, F \rangle$, where D is a domain determining the individuals that exist within the model, and F is a function which provides for each predicate within the semantic representation language a basic semantic value. As a basic example, a set-theoretic model structure for a first-order predicate calculus semantic representation language, might define D to be a set of individual entities, while F would for each one-place predicate assign a set of individuals from D, for each two-place predicate a set of pairs of individuals from D, and so on for higher-arity predicates. For example, a particular model M_1 for this model structure might have $D = \{e_1, e_2, e_3\}$ and assuming a first-order predicate calculus semantic representation language with two predicates, a one-place predicate farmer and a two place predicate love, then we could have $F(\mathsf{farmer}) = \{e_1, e_2, e_3\}$ and $F(\mathsf{love}) = \{\langle e_1, e_2 \rangle, \langle e_2, e_3 \rangle, \langle e_3, e_1 \rangle\}$.

I will begin by discussing what sort of model domain D is required for the analysis of plural denotational structures and follow this with a discussion on the variety of verbal readings that seem available when plural quantifiers interact with verbal predicates. The discussion will bring in to question what sort of information we can expect the function F to provide for arbitrary models.

2.3.1 Model Structure

The first question that has to be addressed when dealing with plural denotations is how collections of individuals and the individuals themselves are to be described within the model structure, and secondly how the two are to be related to each other. Two prominent opposing viewpoints have been given within the literature concerning this question. I will first discuss Godehard Link's lattice-theoretic approach and then look at Fred Landman's set-theoretic counter-proposal[9]

Link (1983; 1984; 1987) proposed that a model-theoretic domain for plurals should be a complete atomic join semilattice, $\langle\langle A, +, \sqsubseteq, AT \rangle, [\![]\!] \rangle$. A is a set partially ordered by $\sqsubseteq$. The join (or summation) operation $+$ takes a non-empty subset $B \subseteq A$ and maps it onto an element of A. Furthermore, a is a minimal element in A iff for every $b \in A : b \sqsubseteq a \rightarrow b = a$. The structure AT, the set of atoms in A, is the set of all minimal elements in A. The symbol $[\![]\!]$ is the semantic interpretation function. Any individual in A is a singular individual if it is in AT otherwise it is

[9] In discussing Link's theory I shall be following Landman's lucid review given in Landman (1989a).

a plural individual. It should be recognised that plural model-theoretic objects within the theory are real **individuals** whose "internal structure" is defined by the lattice partial ordering.

Utilizing this model structure Link can differentiate between verbal predicates which are distributive and collective. A sentence like (46) can be read either distributively or collectively.

(46) John and Bill carry a piano upstairs.

It could be that John and Bill each carried a piano upstairs (distributive) or John and Bill together carried a piano upstairs (collective). The standard assumption within the literature is that when (46) is read in a solely collective manner we would **not** want this to imply that *John carried a piano upstairs* is a truthful statement in this situation although we would wish to accept that John was *involved* in carrying a piano upstairs. The noun phrase *John and Bill* denotes the plural individual derived from the atoms *John* and *Bill*. For a collective reading of (46) we check if this plural individual is in the extension (provided by F) of the verbal predicate carry. For a distributive reading, Link provides a distribution operator which checks that for all minimal (i.e., atomic) individuals within the plural individual for *John and Bill*, that they are within the extension of the verbal predicate for carry. Link (1983) also uses his analysis to handle mass terms which involve the existence of plural individuals which do not contain any minimal elements. Link (1987) has also attempted to conform his analysis to be consistent with generalized quantifier theory.

Both Link and Landman claim individuals and sums of individuals (plural individuals) can not cope with certain constructions containing plural individuals for which the involvement implication of sums (illustrated by the previous example (46)) is false. Following an example by Landman (1989a), assume the pop group Talking Heads consists of the musicians David, Chris, Jerry and Tina. In this situation (47) may be true, although (48) is false because Jerry was ill and didn't perform.

(47) The Talking Heads gave a concert in Holland.

(48) David, Chris, Jerry and Tina gave a concert in Holland.

This contrast can not be coped with in the basic individual/sum framework of Link described so far as both the *Talking Heads* and *David, Chris, Jerry and Tina* would represent the same plural individual either satisfying or dissatisfying both the sentences in (47) and (48). Landman (1989a) describes a further problem for a plural semantics based solely on individuals and sums shown by the following example.

(49) To play this game, the cards below seven and the cards from seven up have to be separated, because we only play with the cards higher than six. (p. 574)

Within Link's basic theory described above the noun phrase *the cards below seven and the cards from seven up* would be constructed from the sum of the two sums of cards below seven and cards from seven up. However, this sum is just the plural individual denoting the cards from the entire pack. This would mean that within the theory both (50) and (51) (amongst others) denoted the same plural individual.

(50) the cards below seven and the cards from seven up.

(51) the cards below ten and the cards from ten up.

Obviously, this is going to give us a wrong interpretation to (49). What is needed is a formulation containing group individuals which are somehow distinct from the individuals which go to make up the group.

Link extends his lattice structure to a new join semilattice which contains two sub-structures, each of which is a complete atomic semilattice. The new structure has two types of atom, **pure** and **impure**. The pure atoms are the atoms from the old structure, while the impure atoms are the new group individuals. One of the two complete atomic semilattices is the structure of pure sums derived from only the pure atoms. The other complete atomic semilattice contains the structure of impure sums derived from only the impure atoms. The whole structure also contains mixed sums containing both pure and impure atoms. The formal structure is $\langle\langle A, \sqsubseteq, +, AT_p, AT_I, \uparrow, \downarrow\rangle, [[]]\rangle$. The symbols, A, $\sqsubseteq$, $+$ and $[[]]$ are defined as before. AT_p is the set of pure atoms and AT_I is the set of impure atoms. The $\uparrow$ and $\downarrow$ operations allow groups to be formed from sums and the elements of a group to be defined. These are defined formally below[10]

- $\uparrow$ is a function from **pure sums** to **impure atoms**.
- $\downarrow$ is a function from **impure atoms** to **pure sums**.

A frequently discussed problems concerns contrasting examples involving the noun *committee*. Even if a committee A and a Committee B have the same members we would not want to conclude (53) from (52).

(52) Committee A paid an official visit to South Africa.

(53) Committee B paid an official visit to South Africa.

Within Link's theory both committees would denote different impure atoms and therefore the implication from (52) to (53) would not be valid. Landman (1989a) provides a illustrative diagram (as shown in figure 2.2) which illustrates the ontological structure of the committees examples. If we assume that both committee A and committee B have John and Bill as members whose denotations are the singular individuals j and b in the model, then figure 2.2 illustrates the fact that the sum of John and Bill $j+b$ is distinct from the impure atoms denoting the committees, C_A and C_B. The $\downarrow$ function relates each committee to the sum $j+b$. Furthermore, the group (impure atom) $\uparrow(j+b)$ derived from the sum $j+b$ is a distinct group in itself.

Against Link's lattice-theoretic approach to plural semantics Landman (1989a; 1989b) has proposed an alternative set-theoretic analysis of plurals. Link's criticisms of a set-theoretic analysis of plurals are summarised and commented on by Landman (1989a, pp. 565-571). Without, reiterating that debate in full, the most powerful argument Link provides is that mass terms can not be handled within a well-founded set-theoretic framework.

Landman shows that some of the possible complete atomic join semilattices definable by the structure $\langle A, +, \sqsubseteq, AT\rangle$ are isomorphic to the set-theoretic domain defined by $\langle S, \subseteq, \cup, AT_s\rangle$. Where AT_s is the set of atoms defined as, $AT_s = \{\{a\} : a \in AT\}$. The singular individuals in AT are replaced by singleton sets in AT_s. The sum operator $+$ is replaced by set-theoretic union $\cup$, while the partial ordering defined by $\sqsubseteq$ is replaced by the set-theoretic subset operator $\subseteq$. The domain S is defined set-theoretically by the closure of AT_s under the union operator. Sums

[10] I am following the notation used by Landman (1989a) to describe Link's theory.

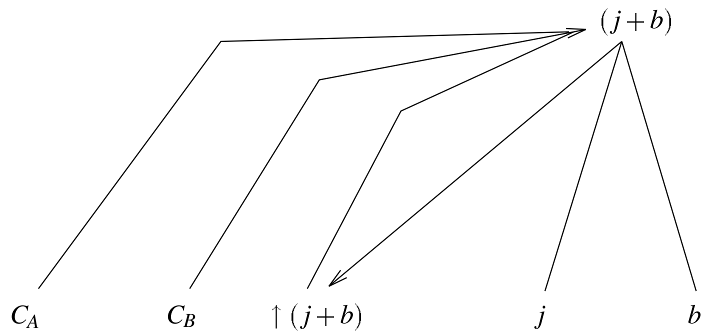


Figure 2.2: The Committees and Individuals Diagram.

(plural individuals) are replaced with sets within *S*. Landman shows that those lattices which are isomorphic to this set-theoretic domain provide reasonable restrictions for the description of count terms[11].

Landman then extends his set-theoretic interpretation of plurals to handle groups. Groups are defined by creating higher-order set-theoretic denotations. For instance the group interpretation of the noun phrase *Talking Heads* will denote the set $\{\{d,c,j,t\}\}$, where d,c,j,t are the individuals who are members of this group. The phrase *David, Chris, Jerry and Tina* meanwhile will denote the set $\{d,c,j,t\}$. Landman explains the distinction between the two sets as follows,

> If the set $\{d,c,j,t\}$ has a certain property like **be pop stars**, there is no reason why the set $\{\{d,c,j,t\}\}$ should inherit that property; vice versa, we can give $\{\{d,c,j,t\}\}$ the property of **being a pop group** without distributing this to the individual members. Also, we can distinguish the sense in which the **Talking Heads gave a concert in Holland** requires the involvement of all the individual Talking Heads from the sense in which there is no such requirement by claiming that in the first case we predicate a property of the sum $\{d,c,j,t\}$, but in the second case of the group $\{\{d,c,j,t\}\}$[12].

Landman's treatment of groups has some limitations over Link's. It can not handle *implicit* groups as observed within the *committees* examples (52) and (53). Landman's semantics only provides the group derived from the sum of the individual members of the committees, i.e., $\uparrow(j+b)$. However, Landman claims similar problems occur for *explicit* groups which Link's theory can not correctly handle either. Landman's arguments are based on triples like (54) to (56).

(54) The judges are on strike.

(55) The judges are the hangmen.

(56) The hangmen are on strike.

[11] See Landman (1989a, pp. 569-570).

[12] Landman (1989a, p. 584).

Landman argues that a person condemned to death would be foolish to use (55) to conclude (56). He views the difficultly to be an aspect of intensionality and provides a solution based on an intensional treatment of properties. Under his interpretation both implicit and explicit groups are involved in these intensional difficulties. However, I believe the problem is only apparent when we wish to obtain intensional analyses for explicit groups. It is only then that an extensional set-theoretic semantics is forced into a corner under the attack of examples such as (54) to (56). For implicit groups (such as with *committees*), an extensional set-theoretic semantics could choose to provide unique individuals for each such group (following Link's lattice-theoretic solution) but fail to enforce the specification of information determining (within the model structure) what individuals are members of each implicit group. Unlike Landman and Link, I see no reason why for implicit groups the model structure must specify the membership of these groups. The problems involved in the committee examples (52) and (53) only become apparent if we force the model structure to always explicitly provide information concerning group membership instead of assuming this information should be available (or not) within the extensions a model provides to appropriate predicates (in F). In other words, it should be possible to test the truth-conditionality of the sentences in (57) and (58) without assuming this information is also enforced within the model structure (i.e., the lattice-theoretic or set-theoretic denotation of implicit groups).

(57) Committee A has the same members as Committee B.

(58) John and Bill are members of committee A.

Following this line of analysis, the committee problem shown in (52) and (53) is actually different from the problem highlighted in (54) to (56). Within an extensional set-theoretic semantics in which implicit groups are treated (possibly naively) as singular individuals, an extensional analysis can solve the committee problem while treating the conclusion in (56) as valid. Only, if we go beyond an extensional analysis and look for intensional readings of sentences can we provide an analysis for the judges/hangmen distinction.

The predictions made so far seem to suggest that groups are essentially an intensional phenomena. Landman's only other argument, apart from the implicit/explicit group distinction, involves discourses like (49), repeated below.

(59) To play this game, the cards below seven and the cards from seven up have to be separated, because we only play with the cards higher than six.

But Landman admits the following[13],

> One might think that this example is somewhat dubious, because it involves the symmetric predicate **separate**. One might claim that (59) should somehow be derived from (60).
>
> (60) The cards below seven are separated from the cards from seven up.

However, he then discusses an example containing the relational adjective *different* (from Link (1984)) and claims no such derivation is possible here.

(61) The men and the women who were married still had to sleep in different dorms.

[13] Landman p. 575 (1989a).

He also cites examples containing reciprocals (like (62)) as further evidence for groups.

(62) The Leitches and the Latches hate each other

Interestingly enough, constructions with *each other* and *different* are handled within a uniform framework by Moltmann (1992). Moltmann provides an extensional set-theoretic semantic framework in which conjoined noun phrases such as *the men and the women* are treated as ordinary sums. Schwarzchild (1992) similarly rejects the groups approach as a solution to the difficulties described and following Moltmann views the difficulty to lie within the analysis of relational adjectives and reciprocals. These examples are handled by complicating the semantic interpretation provided to the relational adjectives *each other* and *different*. The added complexity of groups is not required.

I have covered all of Landman's arguments for groups and have shown that the examples cited either do not require the model-theoretic domain for an extensional semantics to be extended to contain group individuals or those examples that do require group individuals can be either handled by singular individuals or require the semantics to be pushed past the extensional domain into the intensional domain.

2.3.2 Plurality and Readings

Irrespective of the type of model-theoretic domain used to analyse plural constructions, their interaction with verbal predicates allows a multitude of different sentential readings to become realisable. There has been extensive debates within the literature concerning which readings exist. Within this debate a subsidiary discussion concerns the location of the ambiguity to exist either within the analysis of the particular noun phrases or within the analysis of the verb. Apart from these two possibilities, some theorists such as van der Does (1991) propose that the best strategy is "if there are no overt clues to the contrary, one should grant sentences a single weakest meaning which encompasses the others"[14].

Sentences containing numeral determiners seem to allow a variety of readings. Therefore, I shall use the sentence in (63) as a basic example from which the different readings will be discussed.

(63) Four men lifted two tables.

The most basic split is between distributive and collective readings. Assuming both noun phrases are read symmetrically with the same type of reading, a standard distributive reading for (63) (with subject wide quantifier scoping) would require that each of four men lifted two tables. A standard collective reading would require that a collection of four men lifted a collection of two tables. For collective readings it is assumed that the verb is predicated on the collection of individuals, i.e., the collection of men *together* lifted the tables.

I shall follow van der Does' (1991) excellent exposition in discussing the different readings and in particular I shall assume a set-theoretic model structure such as the structure provided by Landman (excluding the group interpretations). I shall assume a function AT, such that if D is a set of individuals, then $AT(D) = \{\{d\} | d \in D\}$.

[14] Van de Does (1991, p. 1).

Scha (1981) derives different readings by placing the ambiguity within the analysis of the determiner. A numeral determiner **(exactly) n** has two collective (C_1 and C_2) and one distributive (D_1) reading. Under van der Does' notation these readings can be stated as follows.

- C_1 $\lambda X \lambda Y. \exists A \in \{Z \subseteq \cup AT(X) : |Z| = n\} : Y(A)$
- C_2 $\lambda X \lambda Y. |\cup \{Z \subseteq \cup AT(X) : Y(Z)\}| = n$
- D_1 $\lambda X \lambda Y. |\cup \{Z \subseteq \cup AT(X) : Y(Z)\}| = n$

The C_1 reading requires that there is some set of individuals A whose magnitude is n and which satisfies the interpretation. The C_2 reading requires there are sets of individuals Z which satisfy the interpretation and whose union is a set of magnitude n. The D_1 reading requires that there is exactly one set of magnitude n all of whose atoms satisfy the interpretation. Given that each determiner in (63) is three-way ambiguous, there are eighteen possible readings, nine possible readings for each quantifier ordering. Providing a symmetrical distributive reading D_1, D_1 (subject-wide quantifier scope) for (63) produces the following reading.

(64) $|\{d \in [\![\text{man}]\!] : |\{d' \in [\![\text{table}]\!] : [\![Lift]\!](\{d\})(\{d'\})\}| = 2\}| = 4$

The semantics of (64) formalises the standard distributive reading discussed above. The symmetrical collective reading of (63) C_1, C_1 derives the following truth-conditional denotation.

(65) $\exists X \in [\![\text{four men}]\!] \exists Y \in [\![\text{two table}]\!] : [\![Lift]\!](X)(Y)$

This collective reading formalises the standard collective reading given above although it implies there are *at least* four men (in a collection) who lift (a collection) of two tables. The symmetrical collective reading of (63) C_2, C_2 is as follows.

(66) $|\cup \{X \subseteq [\![\text{man}]\!] : |\cup \{Y \subseteq [\![\text{table}]\!] : [\![Lift]\!](X)(Y)\}| = 2\}| = 4$

Scha introduces this alternative reading after considering discourses such as (67) below.

(67) Six boys gather.

A C_1 reading of (67) would require that a collection of six boys gather. However, Scha believes this sentence also satisfies a situation where two collections of three boys gather. Another example which perhaps show more forcefully the requirement for an additional collective reading is given in (68).

(68) John, Bill and Harry wrote operas.

The C_1 reading of the subject noun phrase would require that the collection of John, Bill and Harry wrote operas, while a C_2 reading would allow a situation (disallowed by C_1) where John and Bill wrote an opera and Harry another opera. The C_2 reading checks the magnitude of the *summation* of all the sets that satisfy the verbal predicate. It should be noted that unlike the C_1, C_1 reading the C_2, C_2 of (63) requires there to be exactly four table-lifting men. Van der Does comments upon the unlikely models that satisfy a C_2, C_2 reading of (63).

> It is understood that the number of tables involved may vary from two (object wide scope) to eight (subject wide scope). Instead, it varies here from two (object wide scope) to thirty-two (= the number of collections which can be formed out of four men $\times$ two), or thirty (as before, but with the empty collection excluded)[15].

The other combinations of determiner readings for (63) have varying degrees of acceptability. However, an example which allows the contrast between a D_1, D_1 reading and a D_1, C_1 reading to be illustrated is shown below.

(69) Four cooks bought fifty eggs.

In (69), a D_1, C_1 reading would require that each of the four cooks bought a collection of fifty eggs (maybe in a box), while a D_1, D_1 reading would require that each cook bought each of fifty eggs. Thus, in the D_1, D_1 the buying act is presumed to distribute over each egg, while with the D_1, C_1 reading the buying act is only over the *collection* of eggs not over the individual eggs within that collection.

Link (1983; 1984) provides only the standard collective reading C_1 within his semantics. For distributive readings, his analysis is slightly different. His distributive (numeral) reading of a determiner, D_2, is shown below (using van der Does' formalisation).

- $D_2\ \lambda X \lambda Y \exists Z \subseteq X[|Z| = n \wedge AT(Z) \subseteq Y]$

The D_2 reading differs from the D_1 reading in that it requires at least n individuals to satisfy the reading not exactly n as with D_1. When these numeral determiner semantic definitions are generalised for a wider range of determiners (see van der Does (1991, pp. 21-25)) some of the distinctions between readings found for numeral determiners may disappear or be inapplicable. For instance, for the determiner *every* the D_1 and D_2 readings are identical, while collective readings for *every* seem difficult.

Van der Does (1991) argues for a further two collective readings derived from C_1 and C_2 and in consequence highlights the difficultly about what sort of information can be expected within a model. He provides the following series of sentences.

(70) Richard and Ellen play chess together.

(71) Irma and André play chess together.

(72) Two boys play chess.

Van der Does argues that if (70) and (71) are true within a model then so should (72). He then points out that:

> If *Play chess* holds just of pairs, is collective throughout, none of the numeral denotations introduced thus far will be of any help. The distributive denotations will discard the boys since they count atoms, while the collective denotations are insensitive to mixed collections containing both girls and boys.

[15] Van der Does (1991, p. 9).

He proposes that the two collective readings be given extended variants that check for situations in which the individuals *partake in* some action for which a model might not explicitly determine. These new readings are given below, where C_3 extends C_1 and C_4 extends C_2.

- C_3 $\lambda X \lambda Y |\cup\{Z \subseteq X : Z \subseteq \cup Y\}| = n$
- C_4 $\lambda X \lambda Y \exists Z \subseteq X[|Z| = n \wedge Z \subseteq \cup Y]$

This proposed requirement reopens the question concerning the structure that should be expected within models. Within section 2.3.1, I discussed whether we should expect the model structure itself to express the connection between a group and the individual members of that group. I argued that we should not expect these informational implications within a model structure itself but assume that they may or **may not** be determined within the model-dependent denotations provided to appropriate predicates, such as is-a-member-of. Here, van der Does follows my judgement, in not expecting models to always contain the appropriate information structures to satisfy the implication as shown in (70) to (72). However, he does expect that verbal readings should be available to in some sense *patch up* the under-informed model and thus allow the implication in (70) to (72) to succeed. A similar argument can be seen behind the C_1 and C_2 readings. Within the present discussion we can view the C_2 readings as have being introduced by Scha to patch up models in which the C_1 reading fails, such as (67) where we allowed the sentence *six boys gathered* to be valid within models that only specified that two collections of three boys gathered. The question is whether these inferences are best dealt with by defining different readings or whether actually these are more inline with some sort of commonsense reasoning component. Another possibility would be simply to have C_1 only, and allow sentences like (72) and (67) to fail in some unintuitive models, that do not contain the required assignments in F.

Scha (1981) proposes a further quantificational effect which he terms cumulative quantification which he believes can be found in sentences such as (73) below.

(73) 600 Dutch firms have 5000 American computers.

He believes this sentence has a reading which can be paraphrased as: *The number of Dutch firms which have an American computer is 600, and the number of American computers possessed by a Dutch firm is 5000.*

There has been a lot of discussion (van der Does, 1991; Lonning, 1991; Scha, 1991) concerning whether all these different readings are realisable and whether actually a smaller subset will cover all the data. As this discussion takes us too far from the concerns of anaphoric processing I will not enter into this debate. The important aspect for any anaphoric semantic theory is that it can incorporate the wide variety readings that have been proposed.

2.4 Constraints on Anaphoric Reference

It is usually the case that given a set of antecedents made available by a discourse, not all antecedents are available or allowable as referents for pronominal anaphors in all possible continuations of that discourse. This section will review some of the forms of constraint that have been proposed to limit the available antecedents for an anaphor.

<table>
<tr><th colspan="4">Anaphoric Processing</th></tr>
<tr><td>Derivation</td><td colspan="3">Constraint</td></tr>
<tr><td></td><td>Accessibility</td><td>Satisfiability</td><td>Resolution</td></tr>
</table>

Figure 2.3: The different constraint areas for anaphoric processing

Many theories severely limit the range of antecedents they *derive* and thus implicitly place constraints on the available antecedents for a prospective anaphor. In general, given the derivation of antecedents, anaphoric constraints can be broken into the three types given below.

1. Structural Accessibility.
2. Satisfiability.
3. Resolution.

Structural accessibility constraints can be found in both syntactic and semantic theories. They depend on some structural characteristics of the syntactic or semantic representations in order to determine whether an anaphor can reference a particular antecedent. Satisfiability constraints utilize the particular interpretation given to a sentence against a particular model in order to check for the validity of anaphoric references. Resolution techniques generally attempt to use world knowledge and inference mechanisms to determine the best candidate antecedent for a particular anaphor. In the following two sections I will look at structural constraints and satisfiability constraints. Resolution constraints will not be discussed as these generally require the use of world-knowledge and methods of commonsense reasoning which fall outside the present work; see, for example, Grosz (1986), Sidner (1983), Hobbs (1986).

2.4.1 Structural Constraints

Structural constraints on anaphoric accessibility have been utilized in both the syntactic and the semantic domain. Structural *syntactic* constraints are usually based around information supplied by some constituent structure representation, such as a parse tree. To this extent, they depend on the particular brand of syntactic theory being used. However, a phrase structure analysis is common to many approaches. A popular syntactic structural constraint based on phrase structure parse trees is the c-command constraint of Reinhart (Reinhart, 1976; Reinhart, 1983). The c-command rule as defined by Reinhart (1983, p. 41) is given below.

- A node A c-commands node B if the branching node α_1 most immediately dominating A either dominates B or is immediately dominated by a node α_2 which dominates B, and α_2 is of the same category as α_1[16].

This rule (and variants of it) have been used by Reinhart herself and others, including Chomsky (1981) within his Government and Binding framework, to formulate constraints to restrict the

[16]A branching node is a node with more than one daughter, while a node A dominates a node B if A is on the path from B to the root of the parse tree. If a node *A* c-commands a node *B* then *B* is in the c-command domain of *A*.

possible co-indexing of nominal phrases within the grammar. However, these constraints invariably fail in a manner which is difficult to remedy. For example, Reinhart (1983, p. 122) provides a constraint to restrict possible anaphor-antecedent relations involving quantified noun phrases.

- Quantified NPs...can have anaphoric relations only with pronouns in their c-command domain.

This allows her to correctly disallow an interpretation in which the anaphor *he* is co-indexed with the antecedent phrase *an applicant* in (74).

(74) If he turns up, tell an applicant to wait outside.

However, the rule also disallows the following sentences.

(75) I talked with every student about his problems.

(76) That people hate him disturbs every president.

Carter (1987, p. 63) comments that Reinhart's rule "can only be repaired, if at all, by ad-hoc modifications to the theory". These examples and Carter's comment highlight two problems common to structural approaches to anaphoric constraint.

1. They provide rigid forms of restriction.

2. They depend on structural representations whose purpose is not limited to satisfying the constraint mechanism itself, thereby limiting the possibility for changes to the constraint due to its dependence on the particular structural representation.

Structural accounts through their rigidity allow easy discussion of the pertinent data, but this does not mean that a rigid structural account is the best method for constraining possible anaphor-antecedent relations. The second point highlights the fact that structural constraints usually utilize structural representations that are independently motivated and have other uses over and above their use as a representation to base a structural restriction. For a comprehensive discussion of syntactic constraints on anaphora, see Dalrymple (1993).

Two semantic theories which rely heavily on structurally derived constraints are Discourse Representation Theory (DRT) (Kamp, 1981; Kamp & Reyle, 1993) and Dynamic Predicate Logic (DPL) (Groenendijk & Stokhof, 1991b). I will look informally at Kamp's structural constraint in DRT. DRT derives, through the analysis of a discourse, a level of semantic representation termed Discourse Representation Structure (DRS). This structure is utilized directly to ascertain whether a particular anaphor can refer to a particular antecedent. Antecedents are represented by variables whose relative hierarchical position within the level of DRS with respect to a prospective anaphor determines the validity of the proposed reference. The structural constraint in DRT correctly disallows the following discourses, where the pronominal anaphor *it* is taken to refer to the donkeys discussed in the first sentences of these discourses.

(77) Every farmer who owns a donkey beats it. ?It brays in distress.

(78) Joe doesn't own a donkey. ?It lives in a field.

Within DRT, the analysis of universal quantification and negation both result in the construction of subordinate DRSs. Sentences extending a discourse always extend the top-level DRS. Thus, for the analysis of the second sentence of (77) the anaphor *it* needs to be translated as the variable associated with the antecedent phrase for *a donkey*. However, this variable is inaccessible and thus the DRS construction is blocked and the discourse is unacceptable. Like the c-command constraint of Reinhart the structural constraint of Kamp is too rigid and disallows some so called subordination examples. Roberts (1987; 1989) provides examples of modal subordination from investigation of pairs of discourses such as the following[17].

(79) If John bought a book, he'll be reading it by now. ?It's a murder mystery.

(80) If John bought a book, he'll be reading it by now. It'll be a murder mystery.

The continuation of the modal effect of the first sentence in (80) allows the second to be acceptable. Examples exist though in which modal effects seem not to be present.

(81) Every farmer owns a donkey. He uses it in his fields.

(82) Every chess set comes with a spare pawn. It is taped to the bottom of the box.

(83) Every rice-grower in Korea owns a wooden cart. He uses it when he harvests the crop. (Sells, 1986, p. 436)

Roberts derives an analysis of subordination examples within a DRT framework by allowing previous DRS constraints to be accommodated (basically, copied) so as to resituate them within the semantic representation in an accessible position. DRT and DPL will be discussed in detail in the next chapter.

2.4.2 Satisfiability Constraints

Satisfiability constraints concern the imposition of constraint mechanisms on the denotational structures provided to antecedents and anaphors through the interpretational process. For anaphoric purposes semantic number agreement is the most obvious candidate for constraint purposes and has been advocated by Elworthy (1993). At first sight it would seem that anaphor and antecedent phrases must agree in *syntactic* number, as shown below.

(84) Every soprano thinks she is the greatest singer.

(85) ?Every soprano thinks they are the greatest singers.

However, in inter-sentential cases this seems not to be the case.

(86) Every farmer owns a donkey. They beat them to make them work harder.

In (86), the plural pronoun is allowed due to the expectation that the phrase *every farmer* is concerned with more than one farmer. For cases involving only a single anaphoric pronoun, this is more difficult.

[17] In fact, there is a footnote in Cooper (1979, p. 73) where the author attributes Lauri Karttunen as pointing out the existence of acceptable discourses similar to the subordination examples presented here.

(87) Every soprano thinks she is the greatest singer. ?She milks the applause for all it is worth.

(88) Every soprano thinks she is the greatest singer. They milk the applause for all it is worth.

Although in other cases, a singular pronoun is valid.

(89) Every boy comes in. He sits down. He takes out his pen and begins to write.

Semantic number agreement can help augment purely syntactic number agreement checks between anaphor and antecedent.

However, for syntactically plural antecedent phrases, syntactic agreement seems to be required.

(90) ?Most sopranos think she is the greatest singer.

(91) Most sopranos think they are the greatest singers.

(92) Most farmers own a donkey. ?He beats it.

(93) Most farmers own a donkey. They beat it/them.

(94) Five farmers own a donkey. ?He beats it.

(95) Five farmers own a donkey They beat it/them.

However, in certain forms of discourse such as in jokes the restriction may be blatantly ignored for comic effect, as the example below illustrates.

(96) I see my fan club are in tonight. She's sitting in the front row.

Semantic number agreement depends on the interpretation of a discourse with respect to a particular model. The important aspect now, is that the antecedent itself (rather than the syntactic number of the antecedent phrase) agrees in number with the expected number required by the anaphor. Thus, a plural anaphor, such as *they*, requires that its antecedent is some collection of one or more individuals (from the model). Singular anaphors, such as *she*, require that the antecedent be a single individual **or** the antecedent is a collection of individuals that can be individuated via the process of some **bound** anaphor-antecedent relation.

Chapter 3

Semantic Anaphoric Theories

Richard Montague (1974a; 1974b) brought the study of the semantics of natural language within a secure formal (model-theoretic) grounding. Many of the recent post-Montagovian theories attempting to extend the coverage of Montague Semantics[1] look at extending the single sentential limitation to allow the coverage of multi-sentential discourses. Robin Cooper (1979) looks at the problems of discourse anaphora (and donkey sentences in particular) within a Montagovian framework using an E-type analysis of pronouns. Hans Kamp's Discourse Representation Theory (DRT) (1981) presents an alternative non-Montagovian theory of discourse and anaphora that also investigates donkey sentences. Around the same time, Irene Heim developed a philosophically and empirically (though not formally) similar account of discourse anaphora in her File Change Semantics (FCS) (Heim, 1982; Heim, 1988). From the mid 1980s, a series of Montagovian approaches to discourse appeared covering the same empirical ground as DRT. Dynamic Predicate Logic (DPL) (Groenendijk & Stokhof, 1990b; Groenendijk & Stokhof, 1991b) attempts to provide a Montagovian-based account of discourse anaphora which retained the strict notion of compositionality displayed within Montague Semantics and rejected by Kamp in DRT. Dynamic Montague Grammar (DMG) (Groenendijk & Stokhof, 1990a; Groenendijk & Stokhof, 1991a) extends the treatment in DPL providing a fully compositional account from syntax to semantic interpretation utilizing an intensional logic, related in the manner of its formal exposition to the Intensional Logic found in Montague Semantics. Dynamic Type Theory (DTT) (Chierchia, 1991; Chierchia, 1992a) attempts to provide a dynamic logic approach utilizing generalized quantifiers that covered the same ground as DRT/DPL as well as providing a more complex analysis of adverbs of quantification and utilizing an E-type account of anaphora. Van den Berg (1990) extends the basis of the DPL formalism to provide a basic analysis of plurals in his Dynamic Predicate Logic for Plurals (DPLP). Likewise, Kamp and Reyle (1993) extend DRT to handle plural constructions as well as tense and aspect.

The above theories provide a prominent though not complete selection of the formal semantic theories which have specifically covered discourse anaphora. These theories can be partitioned as to whether they provide bound or E-type anaphor-antecedent relations and as to whether they are Montagovian with respect to compositionality and representational issues. The groupings under this partition are reflected in table 3.1. This chapter will provide an in-depth discussion of three prominent theories, one from each class of theories under the partition in table 3.1. Specifically, I

[1] For the definitive introduction see Dowty, Wall and Peters (1981).

	Bound	*E-type*
Montagovian	DPL,DMG,DPLP	DTT,Cooper
Non-Montagovian	DRT,FCS	

Table 3.1: Grouping anaphoric theories with respect to anaphor-antecedent relation and Montagovian philosophy.

shall be discussing Cooper's theory of discourse anaphora, Discourse Representation Theory and Dynamic Predicate Logic. The discussion will concentrate on three sets of comparable theoretical objectives which I believe a semantic theory or framework of anaphora must provide adequate solutions. These objectives are listed below.

1. **Truth and Information**
   (a) Truth Conditions.
   (b) Information.
2. **Pronouns, Verbal Relations and Quantification**
   (a) Referential and bound anaphor-antecedent relations.
   (b) Analysis of verbal relations.
   (c) Analysis of quantification.
3. **Anaphoric Constraints**
   (a) Derivation of antecedents.
   (b) Reference constraining of antecedents.

Before beginning with the detailed discussion of each theory, I will present a brief case of why any anaphoric semantics should be analysed through its ability to handle these objectives.

Determining the truth conditions for declarative sentences (and eventually groups of declarative sentences ordered into a discourse) has been an important goal for semantics since the seminal work of Gottlob Frege (1892; 1952). Montague's work allowed, for the first time, a formal exposition determining the truth conditions of a natural language (English) fragment to be given. However, since formal semanticists have ventured into the realms of discourse and discourse anaphora, the (anaphoric) *information* a discourse makes available has become of similar importance. For example, given the first two sentences of the discourse given in (97), not only do we wish to know under which particular situation (model) it is true that Jack owns a donkey and Jane a horse but that there are individuals Jack, Jane, a horse and a donkey which are related in certain ways and possibly open to later anaphoric reference.

(97) Jack owns a donkey. Jane owns a horse. They beat them.

Indeed, this latter information is required when we come to deal with the third sentence within (97). We need to utilize the various pieces of information derived from the previous discourse to correctly handle this sentence. For a **referential** anaphor-antecedent reading, we need to collect the informational entities for Jack and Jane together to form the appropriate referent for the interpretation of the anaphor *they*. Similarly, we need to collect the entities for the donkey and the horse that Jack and Jane own respectively. However, we do not wish or need to use the information concerning the ownership relationships between these people and their animals. They might beat each other's animals, for instance. For a **bound** anaphor-antecedent reading we must keep the ownership information and use it to ensure that each person only beats the animal they own. By efficiently using the different pieces of information a discourse makes available anaphoric sentences can be provided with the appropriate truth conditions. How different theories handle the interaction between truth conditions and discourse information will be an important aspect to consider when providing an appraisal.

The importance of handling both bound and referential anaphor-antecedent relations within an anaphoric semantics goes without saying. However, it will be important to check whether a particular theory has the flexibility to provide both forms of anaphor-antecedent relation in a variety of discourse situations. Furthermore, the discussions of the previous chapter have shown that both weak (indefinite lazy) and strong (universal) bound anaphor-antecedent relations exist, as well as readings derived by applying uniqueness constraints to verbal relations in collaboration with either a weak or strong bound anaphor-antecedent relation (i.e. the unique anaphor and unique antecedent readings). It has been already shown that the analysis of verbal relations is related to the determination of anaphor-antecedent relations, thus increasing the importance of investigating how a theory tackles these two objectives. In many theories the analysis of verbal relations is also intricately tied up with the analysis of quantification: a relationship that may be traced back to the influence of predicate calculus as the language in which to represent semantic information. Within first-order predicate calculus the interpretation of predicates (the structures to which lexical verbs are translated into) is very basic . For example, a truth-conditional model-theoretic interpretation of a two place predicate P, is provided below, given a model structure $\langle D, F\rangle$, $[\![]\!]$ is the semantic interpretation function, x and y are variables and g is an assignment function from variables to individuals in D.

(98) $[\![P(x,y)]\!]^{M,g}$ = **True** if $\langle g(x), g(y)\rangle \in F(P)$, **False** otherwise.

In (98), we check if the function g assigns appropriate individuals to the variables to satisfy the predicate. The main work within first-order predicate calculus is carried out by the quantifiers. The interpretation of a universally quantified formula is given below, where x is a variable and ϕ is a formula of first-order predicate calculus. The precise nature of a formula is not of concern here, other than to say that it will probably be defined recursively as consisting of some legal combination of predicates, logical connectives and other quantified formulas.

(99) $[\![\forall x\phi]\!]^{M,g}$ = **True** iff for every value assignment g' such that g' is *exactly* like g *except* possibly for the value it assigns to x $[\![\phi]\!]^{M,g'}$ is **True**, otherwise **False**.

The quantifiers manipulate the use of the assignment function g, which determines drastically how predicates are interpreted within (98). In essence, the interpretation of quantifiers (or more strictly quantified formulas) within first-order predicate calculus carries out two different linguistic tasks.

1. The determination of how quantificational effects are correctly handled.

2. The determination of the particular verbal reading.

For first-order predicate calculus the first of these tasks overrides the second as the calculus' use (within natural language semantics) is restricted to the interpretation of (syntactically) singular noun phrases and distributively read verbal predicates. Within this restricted domain the entire workload can be placed within the interpretation of the quantifiers. However, when the full complexity of plural noun phrases is investigated along with the variety of verbal readings that seem to exist, the question arises as to whether quantifiers can still handle both these tasks. The last chapter has shown that a debate exists as to whether the semantic complexities found within the analysis of plural discourse should be handled within the semantics of noun phrases (quantifiers) or the semantics of verbs (verbal predicates), or both. Therefore, the investigation of how a particular theory handles verbal relations will invariably involve the discussion of how that theory deals with quantification.

The final pair of objectives coming under the title of *Anaphoric Constraints*, were largely covered within section 2.4 of the last chapter. It is important to determine which particular ways (from those discussed within section 2.4) a theory constrains anaphoric reference. In particular, one common distinction found within many theories was highlighted by the comparable objectives given: that of (i) deriving antecedents and (ii) constraining those derived antecedents.

After discussing the three theories, I will provide a less detailed discussion of some other theories that promote interesting analyses. Finally, I will end this chapter with a discussion on the contentious issue of representation and compositionality.

3.1 Cooper's Theory

Robin Cooper (1979) provides "an enrichment of the semantic treatment of pronouns proposed by Richard Montague"[2] that in his opinion allows a more intuitive handling of the pronouns found within quantified donkey sentences. Cooper utilizes a slight extension to the original Montague Semantics allowing quantified noun phrases and proper names to denote sets of characteristics. Cooper describes characteristics as below.

> A characteristic is to be regarded as a function from entities to truth values. Thus we may talk of the characteristic of running or the characteristic of loving Mary. The characteristic of loving Mary, for example, will be a function that assigns truth to those entities which love Mary and falsity to those entities which do not love Mary. Thus, we may also think of the characteristic in terms of the set of entities which love Mary, that is, the set of entities to which the characteristic assigns truth. (p. 62)

Given this treatment the proper name *John* and the quantified noun phrase *a man* are translated into Intensional Logic as below.

(100) $\lambda K K(j)$

(101) $\lambda K \exists u[man'(u) \wedge K(u)]$

[2] Cooper (1979, p. 61)

K is a variable over characteristics, j is a constant denoting *John*, and u is a variable over individuals. The expression in (100) denotes the set of characteristics attributable to *John*. Meanwhile, the expression in (101) denotes the set of all characteristics such that at least one man has each characteristic. Under this treatment a pronoun is translated as below.

(102) $\lambda K K(u_i)$: i is a natural number

Montague allows denumerable many variables u_i, and thus denumerable many translations for a pronoun. The expression in (102) denotes the set of characteristics attributable to the (singular) individual identified by the denotation of the free variable u_i. The particular individual denoted by u_i is determined via the context. However, Cooper proposes that pronouns should additionally be translated by the following expression.

(103) $\lambda K \exists x[\forall y[[{}^{\vee}P_i](y) \equiv y = x] \wedge K(x)]$: i is a natural number

Under (103), a pronoun translates as an abstraction over characteristics K for which there is some unique individual x that satisfies the property P_i. A property is a function from possible worlds to characteristics. P_i, like u_i in (102), is assigned its value via the context.

Cooper's intention in defining this new pronominal translation within the intensional logic is to derive an account that follows the E-type proposals of Evans (Evans, 1977; Evans, 1980). As was discussed within section 2.1, E-type pronouns are treated as definite descriptions whose referent is derived from the antecedent. Within Cooper's analysis the referent of the pronoun is derived from the determination of a particular property from the context in which the sentence is being interpreted. This property, for the analysis of donkey sentences, is supposed to coincide with the information derived from the analysis of the antecedent phrase. The translation of the standard quantified donkey sentence in (104) with this framework is given in (105).

(104) Every farmer who owns a donkey beats it.

(105) $\forall u[farmer'(u) \wedge \exists v[donkey'(v) \wedge own'(u,v)] \rightarrow \exists x[\forall y[[{}^{\vee}S(u)](y) \equiv y = x] \wedge beat'(u,x)]]$

We could paraphrase (105) as saying "for any individual, if it is a farmer and there is a donkey which he owns, then there is some unique individual that bears relation S to him and he beats that individual"[3].

Cooper also provides an analysis of simple singular discourse anaphora. He looks at the following discourses.

(106) John looked up. He smiled.

(107) The man looked up. He smiled.

(108) A man looked up. He smiled.

For each discourse he proposes that the sentence *he smiled* be translated into the intensional logic as $smile'(u_0)$ using the standard pronoun translation given in (102). The value of u_0 will be an individual identified by the context. Depending on the individual identified by the context, anaphoric or exophoric anaphor-antecedent relations will be derived. If it happens that the context identifies the individual identified by the interpretation of *John* in (106), *the man* in (107) and *a man* in (108) then anaphoric readings will have been provided.

[3] Adapted from Cooper (1979, p. 84).

3.1.1 Truth and Information

Cooper's theory is couched within the framework of Montague semantics and thus its treatment of truth and information is very much derived from that work. The primary concern of the semantics is to derive the truth-conditions for sentences (in a discourse). No explicit indication is given as to how a discourse, as opposed to isolated sentences, is to be treated formally. Within Montague Semantics, contextual information is provided by indexes to the semantic interpretation function. The description of the model-theoretic treatment given to first-order predicate calculus above contained two indexes only, that of the model and the assignment function, i.e., $[\![]\!]^{M,g}$. However, in general Montague would allow any number of indexes to provide information from the context which was needed to help derive the interpretation of sentences. One guess at how discourses can be handled following the Montague tradition is that both sentences within the discourses (106) to (108) are interpreted against the same indexical context, model and assignment function. If this is what Cooper assumes then the concept of a discourse is very impoverished and is essentially to be treated as a static description of a domain to which different sentences are interpreted. This is in contrast to the dynamic understanding of a discourse that has become popular with the introduction of the many dynamic logics, such as DPL, DMG and DTT. The context, available through indexing, provides the only mechanism for deriving anaphoric discourses as no information is derived or passed by the semantics between the interpretation of each sentence within a discourse. Montague Semantics' truth-conditional primacy does not allow the use of information (e.g., the derivation of possible antecedents) for any other purpose than that for determining the truth-conditions. This is pointed out succinctly by Chierchia (1992a, page 133)

> Roughly put, in a Montague-style semantics, sentences denote truth-values (at an index) and the meaning of sub-sentential components is what they contribute to the determination of sentence values.

Therefore, Cooper's semantics does not attempt to derive and utilize any information concerning antecedents leaving the solution of this task to the context.

3.1.2 Pronouns, Verbal Relations and Quantification

The theory, as has been shown, contains two possible ways of translating pronouns into the intensional logic. Both interpretations crucially depend on the context to identify the antecedent. The theory as a whole only deals with anaphoric discourse containing (syntactically) singular noun phrases. The translation as given by (102) provides a referential anaphor-antecedent relation. This disallows the subordination examples, three of which are repeated below[4]

(109) Every rice-grower in Korea owns a wooden cart. He uses it when he harvests the crop.

(110) If John bought a book, he'll be reading it by now. It'll be a murder mystery.

(111) Every farmer owns a donkey. He beats it.

[4]Cooper in a footnote on p. 73 recognises that some subordination examples exist but he believes they require a treatment utilizing an iterative or habitual operator.

Cooper can not provide the bound inter-sentential anaphor-antecedent relations required by these examples.

His analysis of donkey sentences seems to provide two of the four readings mentioned in section 2.2: the unique antecedent reading and the unique anaphor reading. This is because the semantics requires that the pragmatics (context) supplies a referent that is unique, but the manner in which it is required to be unique is not specified. For the standard quantified donkey sentence of (104), if the individual identified by $S(u)$ is the unique donkey owned by each farmer then the unique antecedent reading is derived. If the individual identified by $S(u)$ is unique is some other way determined by the context then the unique anaphor reading is derived. However, the semantics can not enforce either one of these readings as the determination of $S(u)$ lies within the pragmatics, outside of the semantic theory. It would, therefore, seem more correct to say that Cooper provides a weaker *unique referent* reading in which the semantics simply ensures that the anaphor refers to a unique individual, unique with respect to some contextually determined property.

The intensional logic used by Cooper provides a similar formal treatment of quantification and verbal relations to that of first-order predicate calculus (discussed previously). This implies it also displays the same quantificational bias in the manner in which it deals with these two objectives.

Anaphoric constraints are not discussed explicitly by Cooper.

3.2 Discourse Representation Theory

Discourse Representation Theory (DRT) has gone through two major phases. I shall discuss the original theory first while the most recent version which extends DRT into the areas of plural anaphora, (as well as tense and aspect) is discussed in section 3.2.4.

Hans Kamp's aim in developing the original DRT (1981) was to show that two different prevalent conceptions of meaning could be addressed within a single formal semantic theory. Most model-theoretic semantics had up to this point concentrated on explicitly identifying the truth-conditions for different types of natural language text, as exemplified by the semantics of Montague (1974a; 1974b). The second conception of meaning is described by Kamp as "that which a language user grasps when he understands the words he hears or reads" and is "concerned to articulate the structure of the representations which speakers construct in response to verbal inputs"[5]. Kamp rephrases this distinction in a talk given to the Aristotelian Society (Kamp, 1985) in which he says:

> ...understanding a textual passage is not only a matter of grasping its truth conditions. It is also a matter of grasping the context it provides for what comes next. (pp. 241-242)

DRT's significance is derived from the manner in which it articulates in a formal manner *the context a textual passage provides for what comes next.*

DRT's original analysis concentrates on singular anaphoric discourse and in particular the problems posed by quantified and conditional donkey sentences, which are treated in an identical

[5]Both quotes are from Kamp (1981, 177)

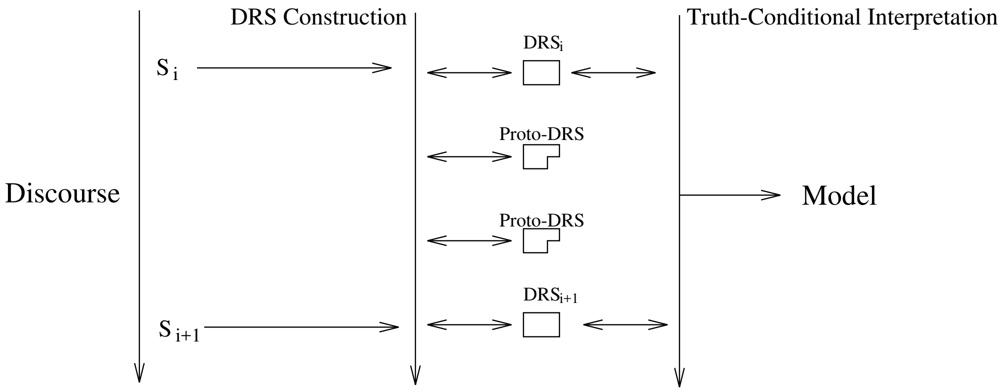


Figure 3.1: The overall structure of DRT.

manner by the theory. The two different conceptions of meaning are formalised by two different operations within the theory. Firstly, discourses are translated into Discourse Representation Structures (DRSs) by the *DRS construction algorithm.* It is this algorithm that formalises the alternative non-truth-conditional conception of meaning. The algorithm is specified by a series of construction rules which work top-down through a syntactic analysis of a sentence and construct a DRS from it. DRT applies these construction rules to each (declarative) sentence in a discourse in the temporal order of the given sentences. Given a sequence of sentences in a discourse $S_1, S_2, ..., S_n$, the algorithm utilizes the DRS K_i derived from the analysis of sentences $S_1, S_2, ..., S_i$ as input to the construction algorithm as it analyses sentence S_{i+1}. For the analysis of sentence S_1 an empty DRS K_0 is used as input. The construction of DRSs essentially covers the objectives 1(a), 3(a), and 3(b) given previously. That is, DRS construction provides a determination of the anaphoric information (antecedents) that are available as well as the constraints existing on those antecedents due to the previous discourse.

The second operation of DRT is to supply an exact truth-conditional interpretation of the DRSs with respect to a model. The DRS structures retain the necessary information onto which a top-down truth-conditional interpretation can be provided. The truth-conditional aspect of DRT essentially covers the objectives 1(a), 2(a), 2(b) and 2(c) given previously. That is, it determines the particular manner in which anaphor-antecedent relations, quantification and verbal relationships constrain the determination of the truth-conditions.

The overall picture of the DRT style of analysis can be visualised as shown in figure 3.1. From this figure, it can be observed that during DRS construction proto-DRSs are created. These are DRSs that are not well-formed and occur during the processing of a sentence by the DRS-construction rules. The truth-conditional interpretation is defined only on well-formed DRSs. An important aspect of DRT that this figure highlights is that both the construction rules *and* the truth-conditional interpretation utilize the DRS form. Thus, the DRSs have to provide the right sort of representational structure so as to allow the proper handling of *all* the objectives outlined earlier. This places a pivotal role on the DRSs which have a dual role to play within the theory. Given that DRT has, since its original creation in 1981, become a general purpose semantic framework in which such diverse subjects as propositional attitudes (Ascher, 1986; Zeevat, 1986) and tense and aspect (Kamp & Reyle, 1993) have been studied, it will be important to ascertain whether a single representational device is best suited to handling both the distinct viewpoints on meaning that were outlined earlier.

I will now give a formal exposition of what constitutes a well-formed DRS. A DRS K is a pair $\langle U_k, Con_k \rangle$, where U_k consists of a set of variables, also known as *discourse referents*, and where CON_k consists of a set of conditions. There can be two types of condition: n-ary place predicates on accessible discourse referents and conditions on other DRSs. Given a DRS $K = \langle U_k, Con_k \rangle$, which has a condition $C_i(J)$, where J is a DRS, the DRS J is called a sub-DRS of K. There is always one DRS which is not a sub-DRS, this DRS is the top-level DRS. Given the above definition, it is possible to formally describe the *empty* DRS used as initial input to the analysis of a discourse. It is simply the DRS $K_0 = \langle \{\}, \{\} \rangle$.

DRSs are usually illustrated in a box notation. For example given the discourse in (112), the construction algorithm produces the DRS shown in (113).

(112) Pedro owns a horse. If he owns a donkey he beats it.

(113)

<table>
<tr><td colspan="3">u v

Pedro(u)
horse(v)
u owns v

<table><tr><td>w x
w = u
donkey(x)
w owns x</td><td>⇒</td><td>y z
z = w
y = x
w beats y</td></tr></table>
</td></tr>
</table>

The discourse referents are displayed along the top of a DRS box. The DRS in (113) shows several (unary and binary) predicates over discourse referents, such as $horse(v)$ and *w owns x*. Also, there is one binary condition on DRSs, $\Rightarrow$, displayed in infix notation within the diagram.

3.2.1 Truth and Information

As has been informally described, the truth-conditional interpretation is defined on the DRSs themselves. The DRSs are treated as partial models which need to be *embedded* within a model. An embedding function verifies a DRS K if each member of U_k is mapped onto an individual in a model and each condition in CON_k is satisfied within that model. If we assume a standard model structure $M = \langle D, F \rangle$, where D is a domain of individuals and F assigns some $X \subseteq D$ to each unary predicate and some $X \subseteq D \times D$ to every binary predicate. Then, given a DRS $K = \langle U_k, Con_k \rangle$ an embedding function f verifies the condition γ in Con_K with respect to M iff:

- γ is of the form $P(x)$, where P is a unary predicate and x is a discourse referent and f maps x onto the individual $a \in D$ and $a \in F(P)$.

- γ is of the form xPy, where P is a binary predicate and x and y are discourse referents and f maps x onto $a \in D$ and y onto $b \in D$ and $\langle a, b \rangle \in F(P)$.

The condition $x = y$, where x and y are discourse referents, is a condition which is verified if f assigns x and y to the same individual in D. The condition $K_1 \Rightarrow K_2$ on DRSs is satisfied by an embedding function f iff for every embedding function g that extends f into U_{k_1} and verifies K_1, there is an embedding function h which extends g into U_{k_2} and verifies K_2. The truth-conditional interpretation looks very similar to the style of interpretation given in Montague semantics for his semantic representation language (intensional logic). However, unlike Montague semantics no direct homomorphism can be provided from syntax to semantic interpretation. The DRS level of representation is non-eliminable.

The derivation and manipulation of anaphoric information is completely handled by the DRS construction algorithm. This means that anaphoric information is dealt with at the representational level. The central locus for anaphoric information is carried by discourse referents. Discourse referents are always assigned to individuals within the model by the truth-conditional interpretation, thus providing the limitation to singular anaphora. A single discourse referent is introduced for each occurrence of a common noun, proper name or pronoun.

3.2.2 Pronouns, Verbal Relations and Quantification

The manner in which the semantics treats anaphor-antecedent relations, verbal readings and quantification is intricately related. Firstly, given the singular nature of the semantics, different verbal readings are not of importance and Kamp simply provides a standard distributive reading to the verbal predicates covered. Two determiners are analysed: the indefinite determiner *a* and the determiner *every*. The DRS construction algorithm treats noun phrases as a whole. Indefinite noun phrases simply introduce a new discourse referent along with a condition specified by the lexical noun associated with the indefinite determiner. Noun phrases headed by the determiner *every* force the introduction of a pair of DRSs predicated with a $\Rightarrow$ condition (i.e., $K_1 \Rightarrow K_2$) within the current DRS. The analysis of the nominal phrase associated with the determiner *every* is placed within the DRS K_1, while the analysis of the associated verb phrase is placed within the DRS K_2. These two situations are illustrated by the DRSs constructed from the sentences in (114) and (115).

(114) A farmer who owns a donkey beats it.

u v
farmer(u) donkey(v) u owns v u beats v

(115) Every farmer who owns a donkey beats it.

u v farmer(u) donkey(v) u owns v	$\Rightarrow$	w w = v u beats w

The effect of these divergent translation mechanisms for each determiner is that indefinites occurring within donkey sentences are located within a different DRS structure to those which occur within more *neutral* situations (as in (114)). Fortunately, this goes hand in hand with the different semantic interpretation required for the determiner *every* and indefinite determiners. The determiner *every* requires universal quantification to be enforced and Kamp's truth-conditional interpretation provides universal quantification to all discourse referents occurring in a DRS K_1 within a DRS $K_1 \Rightarrow K_2$. By supplying universal quantification to **all** discourse referents in K_1, any indefinite noun phrases translated within K_1 also receives universal quantification. This has the effect of deriving the strong (universal) anaphor-antecedent relation for quantified donkey sentences. For example, if we look at (115) again, given the truth-conditions for the conditions of the form $K_1 \Rightarrow K_2$ shown again below in (116), the informal truth-conditional requirements for (115) are given in (117).

(116) The condition $K_1 \Rightarrow K_2$ on DRSs is satisfied by an embedding function f if for every embedding function g that extends f into U_{k_1} and verifies K_1, there is an embedding function h which extends g into U_{k_2} and verifies K_2.

(117) Every u, v such u is a farmer and v is a donkey and u owns v, it must be the case that u beats v.

Kamp's explicit intention is to provide the strong (universal) anaphor-antecedent relation for donkey sentences. We have seen that this is accomplished in conjunction with the quantification provided for the determiner *every*. Outside the effect of the determiner *every*, indefinite noun phrases are essentially translated as free variables, without explicit quantificational effects although the interpretation treats them in an existential manner. However, anaphor-antecedent relations existing outside the effects of the determiner *every* receive an indefinite lazy anaphor-antecedent relation. This is shown in (114), where the truth-conditional requirements informally state that:

(118) There must be some individual u who is a farmer and some individual v who is a donkey such that u owns v and u beats v.

There may be many distinct embedding functions that verify (114), but only one must be found. In particular, some farmer may own and beat half of the 50 donkeys he owns. This situation will satisfy the truth-conditional interpretation of (114) as we simply need to check that a farmer who owns one (or more) donkey beats one (or more) of those donkeys. This is essentially the indefinite lazy anaphor-antecedent relation. Thus, quantificational effects and the derivation of the anaphor-antecedent relations go hand in hand. The situation is summarised in table 3.2.

3.2.3 Constraint Mechanisms

DRT has one major anaphoric constraint mechanism, disregarding gender suitability checks which are also carried out. This mechanism, as was informally discussed in section 2.4.1 of chapter 2, is a form of structural constraint. The structure over which the constraint is defined is the DRS structure. The constraint is applied during the DRS construction algorithm and depends on a notion of accessibility. A discourse referent u is *accessible* from a DRS $K = \langle U_K, CON_K \rangle$ iff:

Discourse Situation	*Quantification given to discourse referents*	*Anaphor-Antecedent Relation*
Within truth-conditional effects of **every**	Universal	Strong (Universal)
Outside truth-conditional effects of **every**	Existential	Weak (Indefinite lazy)

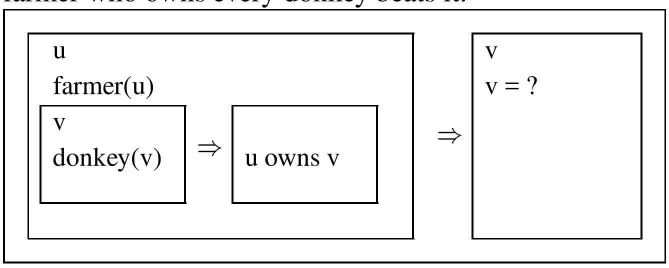
Table 3.2: The relationship between truth-conditions and anaphor-antecedent relations in DRT

1. $u \in U_K$
2. There is a DRSs $K_1 = \langle U_{K_1}, CON_{K_1} \rangle$ and $K_2 \Rightarrow K \in CON_{K_1}$, $K_2 = \langle U_{K_2}, CON_{K_2} \rangle$ and $u \in U_{K_2}$
3. There is a DRS $K_1 = \langle U_{K_1}, CON_{K_1} \rangle$ and for some DRS K_2, either $K_2 \Rightarrow K \in CON_{K_1}$ or $K \Rightarrow K_2 \in CON_{K_1}$ and u is *accessible* from K_1.

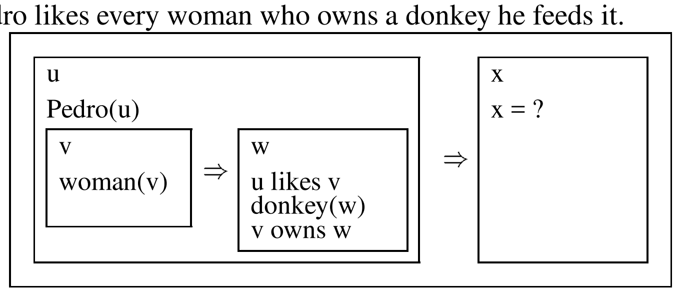
When a pronoun is analysed by the construction algorithm, during the construction of a DRS $K = \langle U_K, CON_K \rangle$ a new discourse referent x is placed within U_K, some *accessible* gender-compatible discourse referent y is chosen and a condition $x = y$ is defined within CON_K.

These constraints allow DRT to correctly disallow certain anaphoric discourses. For example, Kamp (1981, p. 297) looks at the following discourses, whose proto-DRSs are shown alongside.

(119) Every farmer who owns every donkey beats it.



(120) If Pedro likes every woman who owns a donkey he feeds it.



The construction algorithm will not allow the proposed anaphoric references to the discourse referent v for the pronoun *it* in (119) and the discourse referent x for the pronoun *it* in (120) and thus well-formed DRSs can not be constructed and no truth-conditional interpretation is possible. However, I feel sentences like (120) can be made acceptable. For instance, by replacing the verb *likes* in (120) with *visits* (as shown below) the sentence seems much more interpretable.

(121) If Pedro visits every woman who owns a donkey he feeds it

It is interesting that the same complex DRS structures that are derived for handling universal and existential quantification along with universal and indefinite lazy anaphor-antecedent relations are used to control the accessibility of discourse referents. That is, essentially the DRS structure is used for three purposes: quantification, anaphor-antecedent relations and accessibility of discourse referents.

Although not discussed within the original paper, some inter-sentential anaphor-antecedent relations are also blocked. For instance, the discourse in (122) is disallowed, given an attempted anaphoric reference to the discourse referent for *a donkey* by the pronoun *it* in the second sentence.

(122) Every farmer who owns a donkey beats it. *It is old.

However, as has been mentioned before in section 2.4.1, subordination examples exist which contradict the strict structural constraint provided by kamp. Two such subordination examples are shown again below.

(123) Every farmer owns a donkey. He uses it in his fields.

(124) Every chess set comes with a spare pawn. It is taped to the bottom of the box.

Interestingly, even though these inter-sentential constraints were not explicitly discussed in Kamp's original paper they have been used more often to illustrate the theory's anaphoric restrictions than the intra-sentential examples, such as (119) and (120).

3.2.4 DRT and Plural Anaphora

In the early 1990s DRT was extended to handle plural anaphora (Kamp & Reyle, 1990; Kamp & Reyle, 1993). That is, the manipulation of semantically plural discourse referents, which diagrammatically are distinguished from singular discourse referents in DRSs by being displayed in upper case (rather than lower case for singular discourse referents). Formally, following this diagrammatic division, two new conditions are introduced at(x), which means the discourse referent x must denote an individual and, non-at(x), which means the discourse referent x must denote a collection of 2 or more individuals. However, DRT also allows neutral discourse referents (diagrammatically shown using Greek letters) which do not have an attached atomicity condition and can thus range over both singular individuals and plural collections.

The complexity of the theory is dramatically increased from its singular counterpart and I will not discuss all the intricate detailed differences [6]. I will therefore concentrate the discussion on three prominent changes that have had the greatest effect on the appearance and empirical predictions of the theory. These three changes are given below.

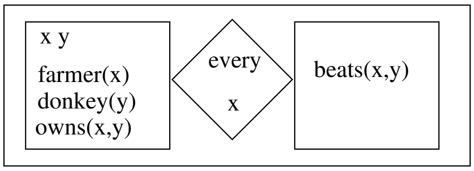
1. The introduction of duplex conditions.
2. Greater discourse referent construction capabilities.
3. The derivation of various verbal readings.

[6]The interested reader can consult pages 305-471 of Kamp and Reyle (1993).

Duplex conditions were developed to allow DRT to utilize generalized quantifiers (Barwise & Cooper, 1981). A sentence whose structure is *Det N VP*, where *Det* is a determiner, *N* is a nominal phrase and *VP* is a verb phrase, is defined within generalized quantifier theory to have verification conditions dependent on the magnitude of the set of individuals that satisfy *N* and the set of individuals that satisfy VP. For the purposes of exposition, let $[\![N]\!]$ be the set of individuals that satisfy a nominal phrase *N*, and let $[\![VP]\!]$ be the set of individuals that satisfy a verb phrase VP. A hypothesis, first described by Barwise and Cooper (1981, pp. 178-179) assumes that we can concentrate our analysis only on $[\![N]\!]$. That is, in determining the satisfiability of a sentence *Det N VP* we can concentrate on some relationship between $[\![N]\!]$ and $[\![VP]\!] \cap [\![N]\!]$. The particular relationship to be tested between these sets depends on the determiner *Det*. For example, the determiner *every* requires that $[\![N]\!] = [\![VP]\!] \cap [\![N]\!]$. Given these changes, the DRS for the standard quantified donkey sentence given below is now shown in (126).

(125) Every farmer who owns a donkey beats it.

(126)



Two DRSs K_1, K_2 connected via a constraint of the form $K_1 \diamond K_2$, where $\diamond = \langle Det, x\rangle$, Det is a determiner and $x \in U_{k_1}$, can be given a generalized quantifier treatment if we ensure that the DRS K_1 relates to the nominal phrase *N* from above and the DRS K_2 relates to the verb phrase VP. We then base the generalised quantifier condition based on *Det* on the individuals x that result from the verification of K_1 and the verification of K_1 extended into K_2. More formally this can be defined below, given the generalised quantifier *R* defined for the determiner *Det*.

(127) The embedding function f verifies $K_1 \diamond K_2$ in a model M (where $\diamond = \langle Det, x\rangle$) iff R holds between the sets A and B, where

1. $A = \{a : \exists g$ that extends $f \cup \{\langle x, a\rangle\}$ and g verifies K_1 in $M\}$
2. $B = \{a : \exists g$ that extends $f \cup \{\langle x, a\rangle\}$ and g verifies K_1 in M and $\exists h$ that extends g and h verifies K_2 in $M\}$

This change to DRT's quantificational analysis not only allows the power of generalized quantifiers to be harnessed but also "solves" the so-called proportion problem that Richards (1984) showed would afflict DRT if it attempted to treat quantifiers such as *most* in a similar way to that of *every*. The problem centres around the fact that the original DRT when interpreting DRSs of the form $K_1 \Rightarrow K_2$ unselectively (universally) quantifies over all discourse referents in K_1, forcing (as has been mentioned above) the simultaneous handling of both the quantificational and anaphoric relationships. If (and it must be stated that Kamp never proposed this) this practise were continued in sentences such as (128) below, incorrect truth-conditions would be defined.

(128) Most farmers who own a donkey beat it.

If the quantificational analysis of (128) unselectively binds both the discourse referent for *farmer* and the discourse referent for *donkey* then we would be checking that most pairs of farmers and donkeys (such that the farmer owns the donkey), it is the case that the farmer beats the donkey. This form of truth-conditional interpretation, however, would stipulate in a situation where there were 100 farmers, 99 who own one donkey and *don't* beat it and one who owns 200 donkeys and beats them all, that (128) was a truthful statement in this situation. The alteration to the use of duplex conditions with their explicit mention of the quantifying variable circumnavigates this possible problem.

However, the type of anaphor-antecedent relation defined by (127) is not the strong (universal) anaphor-antecedent relation but the weak (indefinite lazy) anaphor-antecedent relation. Kamp, therefore reformulates the basic definition of (127) to contain an added condition that enforces a universal anaphor-antecedent relation within DRSs which are duplex conditions. The revised verification condition is given below.

(129) The embedding function f verifies $K_1 \diamond K_2$ in a model M (where $\diamond = \langle Det, x \rangle$) iff R holds between the sets A and B, where

1. $A = \{a : \exists g$ that extends $f \cup \{\langle x, a \rangle\}$ and g verifies K_1 in $M\}$
2. $B = \{a : \exists g$ that extends $f \cup \{\langle x, a \rangle\}$ and g verifies K_1 in M and $\forall j$ that extend $f \cup \{\langle x, a \rangle\}$ and j verifies K_1 in $M \rightarrow \exists h$ that extends j and h verifies K_2 in $M\}$

This more complex condition to enforce a strong anaphor-antecedent relation over the original natural exposition of duplex conditions in which the weak reading came out by itself, might suggest that in some sense the weak reading is the more natural one. However, Kamp's and Reyle's intuitions suggest to them that the strong (universal) anaphor-antecedent reading is the preferred reading and they only provide this reading. The theory has been made considerably more flexible by, at least to some extent, divorcing the quantificational aspect of duplex conditions from the anaphoric considerations. This flexibility, though, is somewhat illusionary as it still requires us to assume that only those linguistic situations in which duplex conditions are required will we require strong (universal) anaphor-antecedent relations.

The second major change found within (Kamp & Reyle, 1993) is that a much wider range of discourse referents can be constructed. The three main operations are given below.

1. Summation.
2. Abstraction.
3. Distribution over Abstraction.

Summation is required to handle the situation where plural anaphors can refer to multiple antecedents. For instance:

(130) John saw Mary. They went to the cinema.

In (130), the anaphor *they* refers to both the antecedent derived from *John* and the antecedent derived from *Mary*. DRT incorporates a summation operator, $\oplus$, that derives the union of a series of discourse referents and places the result in a new discourse referent. The DRS derived by the DRS-construction algorithm from the analysis of (130) is shown below.

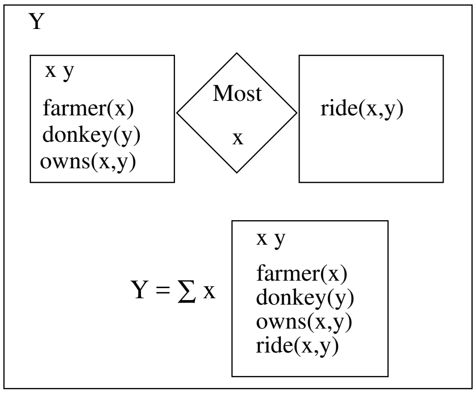
(131)

u v U Z

John(u)
Mary(v)
u saw v
$Z = u \oplus v$
$U = Z$
U went to the cinema

Abstraction is used to derive a new discourse referent which is the union of values derivable from a single discourse referent within a duplex condition. The new discourse referent is placed within the main (top-level) DRS. This, in effect, derives a new *accessible* discourse referent from the summed values of a previously *inaccessible* discourse referent. An example where this is needed in DRT is given below, along with the DRS derived after the analysis of the first sentence.

(132) Most farmers ride a donkey. They prefer this form of transport.



The construction rule for abstraction is written so as to be triggered whenever a duplex condition occurs and to derive all sums of all discourse referents occurring within the duplex condition. Given this, the DRS in 132 is strictly incomplete as there should be an abstracted sum for the discourse referent *y* as well. The process of abstraction is going to considerably swell the size and complexity of DRSs and from (132) it can be seen that each abstracted sum requires the copying of informational constraints existing elsewhere within the main DRS. In this way, abstraction causes the creation of large amounts of duplicated information structures. This could be seen as a (negative) consequence of Kamp and Reyle's wish to manipulate anaphoric information at the representational rather than the denotational level. Added to this however, it is not always clear that the derived sum is the only sum that one might wish to derive from duplex conditions. For instance, consider the following discourse,

(133) Susan found most presents that Bill hid. They were in the garden.

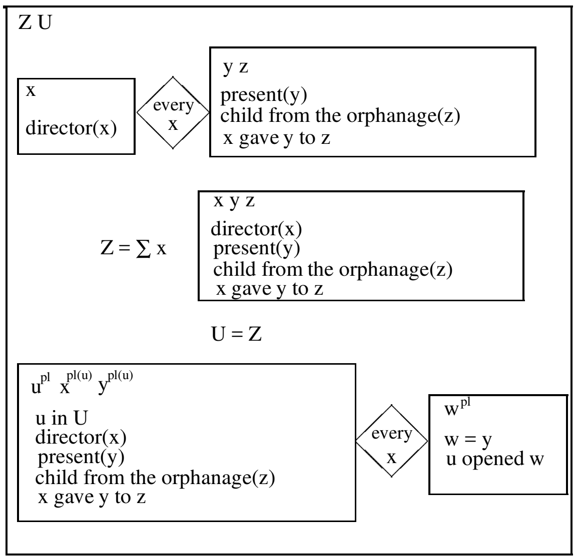
Abstraction over the discourse referent for *presents* will derive the set of *presents that Susan found and Bill hid.* However, the anaphor *they* seems also to have a referent to simply the set of *presents that Bill hid.* In DRT, there is no way of abstracting this information.

Finally, Kamp and Reyle realise that not only can discourse referents be abstracted and summed but the resulting sums can be involved in dependent verbal relations in which the individual members retain their dependencies to each other. The general example he discusses is the following.

(134) Every director gave a present to a child from the Orphanage. They opened them right away. (p. 375)

One of the possible readings for (134) is where the sentence *they opened them right away* can be read as saying that each child opened the present given to him right away. This requires that we enforce the relational dependencies between the discourse referents in (134). Kamp and Reyle derive operations to copy information from an abstracted discourse referent and form a new duplex condition. Further construction rules need to be amended to get around DRT's strict number constraints of discourse referents, as usually the discourse referent for a plural anaphor requires a plural antecedent discourse referent[7]. The resulting DRS for (134) under the required interpretation is given below.

(135)



One problem, first noted by Elworthy (1993, pp. 62-63), follows a similar line to the problem given above for abstraction and is highlighted by the following discourse.

(136) Every farmer owns a donkey. They beat them. They hate them.

A reading in which every farmer beats a donkey owned by some farmer (but not necessarily himself) is not available, as the distribution over abstraction construction rule collects the entire set of constraints pertaining to the abstracted discourse referent. In particular, it is therefore not possible to read the third sentence in (136) as saying that each farmer hates the specific donkey(s) he beats, rather than the one(s) he owns.

With the introduction of the handling of plural noun phrases a varied selection of verbal readings becomes available, as discussed in the previous chapter in section 2.3. However, Kamp and Reyle take the conservative (although understandable) choice of only extending their verbal readings to include the C_1 collective reading of Scha as discussed in chapter 2 in section 2.3.2. For instance, the collective reading of (137) derives a DRS given in (138).

[7]These extensions will be discussed at the end of this section.

(137) Three lawyers hired five cleaners.

(138)

X Y
lawyer*(X)
$|X| = 3$
cleaner*(Y)
$|Y|=5$
X hired Y

However, as one would expect, such a basic analysis of verbal readings leaves the door open to a wide range of readings of discourses which are underivable in DRT. For example, Elworthy (1993, p. 63) shows the following example.

(139) Three boys buy five roses. They like them.

The reading where one boy buys a rose on his own, and then two other boys as a group buy the remaining four roses can not be derived, as essentially this requires the C_2, C_2 reading discussed in the last chapter. Furthermore, distribution over abstraction will not provide the reading where each boy or collection of boys likes the roses he bought which would require some form of "C_2, C_2 reading over abstraction".

Finally, DRT is made more complex by requiring on the one hand a strict division between singular and plural discourse referents and on the other hand an analysis of linguistic data which shows that this strict notion is violated, at least in the manner in which it is expressed within DRT. Therefore, discourse referents introduced by plural noun phrases are given a *pl* superscript which allows them under the auspices of other construction rules to be used as antecedents for plural pronouns. Further additions are needed for dependent pronouns, such as occur in examples such as (134). That is, in cases of distribution over abstraction all ancillary discourse referents other than the principal abstracted discourse referent (which is marked with *pl*) are given $pl(u)$ superscripts where u is the principal abstracted discourse referent.

3.3 Dynamic Predicate Logic

Groenendijk and Stokhof (Groenendijk & Stokhof, 1990b; Groenendijk & Stokhof, 1991b) propose a dynamic semantic interpretation of the language of first-order predicate logic. Their intentions for doing so are clearly stated when they say:

> The resulting system, which will be referred to as 'Dynamic Predicate Logic', is intended as a first step towards a compositional, non-representational theory of discourse semantics. (Groenendijk & Stokhof, 1990b, p. 55)

The principle non-compositional representational theory they methodologically object to is Kamp's DRT, which was discussed in the previous section. Within Dynamic Predicate Logic (DPL) they wish to derive a compositional semantic interpretation for the logical language of first-order predicate logic. Later in Dynamic Montague Grammar (DMG) (Groenendijk & Stokhof, 1990a; Groenendijk & Stokhof, 1991a) they wished to provide a complete compositional analysis from syntax to semantics via the logical language Dynamic Intensional Logic. More specifically they consider DMG as:

> ...one way to 'lift' Dynamic Predicate Logic to a type-theoretic level, the level that is needed to achieve a fully compositional semantic framework along the lines of Montague grammar.

Both theories provide empirically equivalent accounts of discourse anaphora, empirically identical in fact to that of DRT, circa 1981. Furthermore, both accounts equally illustrate the linguistic motivations behind the type of dynamic logical semantics they wish to pursue. DPL is formally more simple and perspicuous compared with DMG whose main achievement is, as stated above, to allow a type-theoretic compositional account in the Montagovian tradition. For these reasons DPL will be discussed extensively here, while DMG will be reviewed briefly in section 3.4.

Within DRT, the radical change required to handle the difficulties of discourse anaphora was a non-eliminable level of representation (DRS) whose existence was not merely subsidiary to the truth-conditional interpretation but provided an alternative and integral level of meaning. Groenendijk and Stokhof also believe a new definition of meaning is required. For them, the meaning of a sentence lies not in its truth-conditions but in its context-change potential. That is, the utterance of a sentence is a transfer from the information state before the utterance of the sentence to an information state after the utterance. In the case of discourse anaphora, the information state can be viewed as containing the antecedents available for anaphora. Thus, the utterance of a sentence changes the available antecedents, possibly adding and/or removing antecedents that are available for later reference by anaphors.

Within DPL, Groenendijk and Stokhof use assignment function pairs to implement this concept of information change potential. An assignment function assigns a value (individual) from the model to every variable used within the logic. To illustrate their analysis, given below is the interpretation of an existentially quantified formula within both first-order predicate logic under a standard static model-theoretic interpretation[8], (140), and Groenendijk and Stokhof's *dynamic* model-theoretic interpretation, (141), where g, g', k are assignment functions, M is a model and ϕ is a formula (of predicate logic in (140) and of DPL in (141)).

(140) $[\![\exists x\phi]\!]^{M,g}$ = **True** iff for some $g' : g'[x]g$ $[\![\phi]\!]^{M,g'}$ is **True**, otherwise **False**.

(141) $[\![\exists x\phi]\!]^{M} = [\langle g,h\rangle \mid \exists g' : g'[x]g \& \langle g',h\rangle \in [\![\phi]\!]^{M}]$

The notation $g'[x]g$ means that the assignment function g' differs from g at most in the value it assigns to the variable x. Under the **static** interpretation we focus the interpretation on all the distinct assignment functions that differ at most in the value g assigns to x and look for one which satisfies the formula ϕ. The formula ϕ might consist of another existentially quantified formula which requires the interpretation to look at all the distinct assignment functions that differ from g' in at most some value it assigns to a particular variable (productively, but not necessarily, some variable other than x). In this way, the interpretation compositionally moves through a given formula looking for assignment functions that satisfy all aspects of the formula. This is irrespective of the original assignment function g used as the overall assignment function for the entire predicate logic formula we are interpreting. From this it is clear the assignment functions that occur during the interpretation and which actually hold interesting information are

[8]Adapted from (Dowty *et al.*, 1981, p. 60).

lost, as we interpret the whole formula with respect to any assignment function g. Groenendijk and Stokhof's **dynamic** variant can be viewed as a way of retaining this lost information and in particular allowing it to be used for the interpretation of subsequent formulas. Their interpretation utilizes input and output assignment functions g and h. The assignment function h is the recipient of all the changes that take place during the interpretation of the formula. The interpretation in (141) can be paraphrased as denoting " all those input-output assignment function pairs $\langle g,h\rangle$ for which there exists an assignment function g' which differs from g in the value it gives to the variable x and the assignment function pair $\langle g',h\rangle$ satisfies ϕ". Groenendijk and Stokhof provide similar dynamic interpretations for all the other constructs of first-order predicate calculus, some of which will be discussed in the subsequent text.

3.3.1 Truth and Information

Within DPL anaphoric information resides within the assignment function pairs which are provided as denotations for formulae. The domain of the assignment functions are individuals within the model. This limits the semantics to the analysis of singular anaphoric information and therefore in consequence to the analysis of syntactically singular determiners, in this case, *every* and *a*. An assignment function within DPL is formally meant to represent an anaphoric information state. Kamp (1990), though, has questioned the validity of describing assignment functions as information states. If they were information states it should be possible to determine which assignment corresponds to "the minimal information state, that in which no information is available"[9]. However, it is hard to see what assignment function(s) could correspond to this information state. Furthermore, assignment functions depend on a particular domain defined by a particular model. But, Kamp argues that information states should not be tied to a particular model or domain. He suggests an alternative treatment in which information states are associated not with assignment functions but a pair $\langle M,f\rangle$ of an assignment function with respect to a particular model. The formulae of DPL would then denote a pair of model and assignment function pairs.

Truth in DPL is defined with respect to a given model, M, and assignment, g. If a formula ϕ which is interpreted with input assignment g and model M has some output assignment h then the formula is $true$. Formally this is stated as follows:

Definition 1: **Truth in DPL**

ϕ is *true* with respect to g in M iff $\exists h : \langle g,h\rangle \in [\![\phi]\!]^M$

3.3.2 Pronouns, Verbal Relations and Quantification

Following DRT (circa 1981), DPL only concerns itself with singular anaphoric reference and thus in consequence limits itself to providing distributive readings for verbal relations. Unlike DRT though, the handling of anaphor-antecedent relations and quantification occurs from the semantic interpretation of two different parts of the semantic representation language. In DRT, both objectives were handled by the overall truth-conditional rule for the two types of DRS structure available. In DPL these two objectives are separated. The analysis of quantified formulas (i.e., $\exists\phi$ or $\forall\phi$) provides the quantificational analysis for the determiners *every* and *a*. I have described

[9] Kamp (1990, p. 111).

the rule for interpreting $\exists\phi$ in (141). Below, is the semantic interpretation rule for $\forall\phi$ which as can be seen translates straightforwardly from the static version given in (143).

(142) $[[\forall x\phi]] = [\langle g,h\rangle \mid h = g \& \forall g' : g'[x]g \Rightarrow \exists m : \langle g',m\rangle \in [[\phi]]]$

(143) $[[\forall x\phi]]^{M,g}$ = **True** iff for every $g' : g'[x]g\ [[\phi]]^{M,g'}$ is **True**, otherwise **False**.

The important difference between (142) and (143) is that DPL requires that we find the assignment functions m which result from the satisfaction of the formula ϕ.

Meanwhile, the analysis of the anaphor-antecedent relations found within donkey sentences is handled by the dynamic semantic interpretation of implication. The translation of the standard quantified donkey sentence in (144) is given in (145).

(144) Every farmer who owns a donkey beats it.

(145) $\forall x[[farmer(x) \wedge \exists y[donkey(y) \wedge own(x,y)]] \rightarrow beat(x,y)]$

The compositional interpretation will require the analysis of the construction $\phi \rightarrow \psi$, where $\phi = [farmer(x) \wedge \exists y[donkey(y) \wedge own(x,y)]$ and $\psi = beat(x,y)$. The interpretation of these formulae is given below.

(146) $[[\phi \rightarrow \psi]] = [\langle g,h\rangle \mid h = g \& \forall k : \langle h,k\rangle \in [[\phi]] \Rightarrow \exists j : \langle k,j\rangle \in [[\psi]]]$

The important element is the universal quantification given to all possible assignment functions k thus forcing universal quantification on the formula ϕ. This effectively forces the variable y in (145) to receive universal quantification, overriding the explicit existential quantification. This provides the universal anaphor-antecedent relation. Although other anaphor-antecedent relations are not discussed by Groenendijk and Stokhof, the interpretation rule for implication could be modified to handle the weak (indefinite lazy) anaphor-antecedent relation. Such a modification is given below in (148), using an adaption derived from their rule for conjunction given in (147).

(147) $[[\phi \wedge \psi]] = [\langle g,h\rangle \mid \exists k : \langle g,k\rangle \in [[\phi]] \& \langle k,h\rangle \in [[\psi]]]$

(148) $[[\phi \rightarrow \psi]] = [\langle g,h\rangle \mid h = g \& \exists k : \langle h,k\rangle \in [[\phi]] \& \exists j\langle k,j\rangle \in [[\psi]]]$

Outside the influence of the implication operator weak (indefinite lazy) anaphor-antecedent relations are provided, similar to DRT. This can be seen by the analysis given to the sentence in (149) which has a translation in DPL given in (150).

(149) A farmer who owns a donkey beats it.

(150) $\exists x[farmer(x) \wedge \exists y[donkey(y) \wedge owns(x,y) \wedge beat(x,y)]]$

The sentence in (149) would be satisfied if at least one farmer beats at least one donkey he owns. A referential reading would require that some farmer owns and beats exactly one donkey. However, I believe that the existence of universal anaphor-antecedent relations doesn't need to go hand in hand with universal quantifiers, in so much as a sentence like (149) should be able to be read with a universal anaphor-antecedent relation. For example, one could imagine a campaign speech by the leader of the farmer's union in Ithaca, containing a section as follows.

(151) We farmers are a proud and traditional workforce. A farmer who owns a donkey beats it. It has always been this way, and I will fight the animal-rights activists who wish to force us to change our traditions.

I believe in this context, (149) is better read with a universal anaphor-antecedent relation applying.

3.3.3 Constraint Mechanisms

Within DPL, no discussion is given as to how an anaphor chooses its antecedent. It is assumed that anaphors are represented by some variable within a DPL formula. Within the examples given by Groenendijk and Stokhof, the "correct" variable is always chosen to allow for the appropriate interpretation. This contrasts with DRT, where as Kamp (1990) comments: "a great deal of fuss is made over just this sort of question"[10]. The anaphoric constraints within DPL occur at the interpretational level and are integrated into the rules that determine the denotational assignment function pairs. In attempting to provide an empirically equivalent theory to that of DRT, the important consideration is to only allow antecedent information derived from indefinite noun phrases to be passed on to further sentences and then only if those indefinite noun phrases are interpreted external to the context of an implication. The context $\phi \rightarrow \psi$ is used by DPL for translating donkey sentences. For instance, both types of standard donkey sentence along with their translations into DPL formula are given below.

(152) Every farmer who owns a donkey beats it.

(153) $\forall x[[farmer(x) \wedge \exists y[donkey(y) \wedge own(x,y)]] \rightarrow beat(x,y)]$

(154) If a farmer owns a donkey he beats it.

(155) $\exists x[farmer(x) \wedge \exists y[donkey(y) \wedge own(x,y)]] \rightarrow beat(x,y)$

Both translations, contain formulae of the form $\phi \rightarrow \psi$ and it is the antecedents derived from ϕ and ψ which must not be accessible from outside this formula if DRT's empirical predictions are to be adhered to. The interpretation of formula of the form $\phi \rightarrow \psi$ is given again below.

(156) $[\![\phi \rightarrow \psi]\!] = [\langle g,h\rangle \mid h = g \& \forall k : \langle h,k\rangle \in [\![\phi]\!] \Rightarrow \exists j : \langle k,j\rangle \in [\![\psi]\!]]$

The important aspect in (156) is the stipulation $h = g$ which ensures that any information that is created during the processing of this formula (as held in the assignment function j) is disregarded with respect to the input-output considerations. A similar stipulation is defined within the analysis of universally quantified formulae, the interpretation of which is shown again below.

(157) $[\![\forall x\phi]\!] = [\langle g,h\rangle \mid h = g \& \forall k : k[x]h \Rightarrow \exists m : \langle k,m\rangle \in [\![\phi]\!]]$

This ensures that antecedents derived from noun phrases of the form *every X* are not available for anaphoric reference outside the scope of the associated universal quantifier to which the determiner *every* is translated. Groenendijk and Stokhof use the terminology *internally* and *externally* dynamic to specify the different types of dynamism available. Implication and universal quantification are internally but **not** externally dynamic. That is they allow the transmission of information (assignment functions) within their compositional analysis but not external to it. Examples of internally and externally dynamic constructions are existential quantification and conjunction.

DPL's anaphoric restrictions enforce similar constraints to those found within DRT. For example, the following anaphoric discourses would not be interpretable within DPL.

(158) If a farmer owns a donkey he beats it. *He hates it.

[10] Kamp (1990, p. 120).

(159) Every farmer owns a donkey. *He beats it.

Given the empirical identity to that of DRT, subordination examples can not be handled. However, the prospects for an appropriate treatment are more difficult than in DRT. In DRT, anaphoric information from the entire discourse is always available, though not always *accessible*. Given this, DRT could be extended with DRS-construction rules which retrieve formerly inaccessible anaphoric information and relocate it into a more accessible DRS. Indeed, this is what was proposed by Roberts (1987; 1989) and incorporated into DRT by Kamp and Reyle with their abstraction mechanism. In DPL, however, information is only available via the assignment functions. The interpretation of implication and universal quantification simply does not allow the existence of this information outside the confines of their local analysis. This would mean that any attempt to handle subordination would require an ambiguity in the dynamics. Thus, externally dynamic forms of universal quantification and implication would be required. Indeed, this is what Groenendijk and Stokhof propose[11]. It can be seen that DPL thus enforces a very black and white picture on the availability of antecedents. If an antecedent is available it is completely available, otherwise if antecedents are derived within externally static contexts outside those contexts they can never be anaphorically retrieved.

3.4 Other Theories

My review of the relevant theories in the area has been particularly selective, although each theory discussed was associated with one of the three main partitions highlighted by table 3.1. Within this section I will look briefly at some of the many other theories that have been proposed. My main criteria for a theory's inclusion is that it allows an opportunity to evaluate an empirical or formal issue not covered by the theories just discussed.

Dynamic Montague Grammar (DMG) (Groenendijk & Stokhof, 1990a; Groenendijk & Stokhof, 1991a) is, as was briefly discussed in section 3.3, the result of one way to lift DPL to the type-theoretic level needed for a fully compositional analysis between syntax and semantics. However, Groenendijk and Stokhof do not use DPL as the intermediate representational language but devise DIL, Dynamic Intensional Logic[12], for this purpose. In doing so, they make a split between ordinary variables, which carry out a similar role to variables in Montague's intensional logic, and discourse markers which carry the dynamic information. Along similar lines there is a split between assignment functions that assign individuals (from the model) to variables and *states* that assign individuals (from the model) to discourse markers. One interesting consequence of the analysis of DPL and DMG is highlighted by David Beaver (1991). He looks at contradictory discourses, such as (160) and (161).

(160) At four o'clock the Hatter had finished his cup of tea, but by four-thirty he still hadn't started it. However, by five o'clock he'd finished it again. (p. 149)

(161) John is self-identical. John is non-self-identical. (p. 150)

Given the assumption that DMG could handle discourse of this complexity, and that the pronoun *it* is treated in a non-sloppy manner to refer to the cup of tea identified in the first sentence

[11] See Groenendijk and Stokhof (1991a, pp. 27-34).

[12] A logical language adapted from work by Janssen (1986).

of (160), DMG's analysis would have the following properties. After the analysis of the first contradictory sentence in (160) (within a model that bears out this contradiction) no individual is available for binding to the anaphor *he*. This seems strange, as language users have no trouble in identifying the required referent. Similarly, in (161) no extension of this discourse could anaphorically refer to the individual identified by *John*. The reason for this problem in DMG (and DPL) is that the truth-conditional analysis and the propagation of information is tightly coupled. Only truth-conditionally valid assignments are available for anaphora. This contrasts with DRT where discourse referents are always in existence, although possibly inaccessible.

Dynamic Type Theory, DTT (Chierchia, 1991; Chierchia, 1992a; Chierchia, 1992b), is a theory of discourse semantics which on the surface is very similar to that proposed in DMG. However, Chierchia provides some extensions and variations to the dynamic logical analysis of DMG. Firstly, he extends his theory to cover generalised quantifiers rather than just existential and universal quantification. The interesting aspects of DTT concerns its analysis of conditional and quantified donkey sentences. I will only discuss the theory's analysis of quantified donkey sentences as conditional donkey sentences will not come under the umbrella of the present work. Chierchia begins by looking at the donkey sentence in (162).

(162) Most farmers who own a donkey beat it.

He sees both strong and weak readings for these sentences, which derive the following truth-conditions.

(163) Weak Reading of (162).
The number of men that own and beat a donkey is greater than the number of men that own but don't beat one.

(164) Strong Reading of (162).
The number of men that own a donkey and beat every donkey they own is greater than the number of men that own a donkey but do not beat every donkey they own.

To accommodate these two readings into his theory he defines the quantifier *most* to be ambiguous between two translations in his logic. He similarly defines ambiguous readings for his other determiners. However, he later shows that the strong-reading translation of determiners derives generalised quantifiers which are not of a conservative nature, as described by Barwise and Cooper (1981) as a universal hypothesis for natural language determiners. Given that most theorists believe this constraint probably valid, Chierchia feels he needs to find an alternative mechanism to derive the strong readings. For this purpose, he looks towards utilizing an E-type analysis. Strong readings are derived by allowing the anaphor to be interpreted as a contextually determined function. For instance, Chierchia shows (165) as an example of a (donkey) sentence that requires a strong (universal) reading.

(165) Every landowner that owned a slave exploited him.

He proposes that the meaning of (165) be represented as the following:

(166) $\forall x[[landowner(x) \wedge \exists y[slave(y) \wedge own(y)(x)]] \rightarrow exploit(x, f(x))]$

If $f(x)$ is a contextually defined function that determines a slave or group of slaves owned by x then Chierchia points out that in the situation where every slave is beaten by its owner then the choice of $f(x)$ does not matter[13]. However, the theory has similar consequences to those found in Cooper's theory discussed in section 3.1 in that he is proposing that semantics is not the right place to determine the operation of these pronouns and that such a job be left for pragmatics.

Certain other theorists (Elworthy, 1993; Zeevat, 1989; Zeevat, 1990) have accused DPL and DMG of being non-traditionalist, as they replace the non-traditional non-eliminable representations in DRT with a non-traditional dynamic interpretation. Zeevat (1990), supplies a static semantics whose particular significance is that existential quantifiers can bind variables outside their scope both to the left and right. This contrasts with DPL where existential quantifiers can only bind variables outside their scope to their right. This addition allows the possibility of handling examples of cataphora. Elworthy (1993) defines a static semantics in which information concerning the content of noun phrase denotations is kept within a *context* with respect to which the interpretation is based. The context provides a denumerable number of *slots*, one for each possible noun phrase that could occur in a discourse. A logic, L(GQA), is defined utilizing this conceptual model. Different anaphor-antecedent relations are provided by giving ambiguous translations to determiners. The theory succeeds in being extremely flexible (albeit with a complex logic). However, the possibility of constraining the flexibility is limited to a series of preference rules which promote the utilization of certain logical translations in L(GQA). Even though these two theories profess to be more traditional in nature, Zeevat has to modify the denotational structures in D_t (the denotations of type t) to include not only the truth values 0 and 1 but also the set of discourse markers. Meanwhile, Elworthy has to define a "non-traditional" notion of a context set against which language is interpreted. It seems, therefore, that any theory, attempting to analyse discourse, at least all the ones I have discussed, requires some extension beyond the basic Montagovian framework. This seems hardly surprising, and although it might have sense, as Groenendijk and Stokhof do, in suggesting one particular type of extension (dynamics) as preferable over another (non-eliminable representations) the most basic requirement we should expect is that theories are perspicuous and cover the empirical data, not that they should be traditional (in the Montagovian sense).

Cooper (1991) provides an E-type account within a situation theoretic framework. The interesting aspect is that whereas in (Cooper, 1979) the identification of the appropriate context to satisfy the reference of the formalised E-type pronoun was left to pragmatic considerations in an unspecified manner, this time the context is specified by possible situations (in the technical situation theoretic sense) and the formalisation takes advantage of this to provide either strong or weak readings dependent on the type of situations existing which describe the context. This prompts Cooper to suggest that the weak and strong reading ambiguity is not an ambiguity in language but an ambiguity in how information is structured in a given context in which an utterance is made.

3.5 Compositionality and Representation

One of the strongest debates between the adherents of the theories discussed in sections 3.1 to 3.4 concerns compositionality and the existence (or not) of non-eliminable representations.

[13]A point he relates, as I did earlier, to the sage plant examples of Kadamon (1990). See section 2.2.3.

The prevalent theories can be split between the representationalist theories, most prominently DRT, and the traditionalist Montagovian-based theories, most prominently DPL, in which a strict notion of compositionality is held to be paramount. The notion of compositionality is generally attributed to Gottlob Frege[14] (1892; 1952) and is exemplified by the phrase “the meaning of the whole is a function of the meaning of the parts and the way they are put together”. However, theories such as DPL take their definition of compositionality from the work of Richard Montague (1974b) in which a much stronger notion of compositionality is defined. Montague required that the syntax of a (disambiguated) language as well as the semantics be defined as algebras and there to be derived a direct homomorphism between structures in the syntactic algebra and structures in the semantic algebra. By the definition of a homomorphism, for each structure in the syntactic algebra there must be a single structure in the semantic algebra to which it corresponds.

A standard technique to simplify the creation of a particular homomorphism between syntax and semantics is to define two separate homomorphisms one from the syntax to some intermediate representational language (e.g., DPL) and another from this language to the semantics proper. As two homomorphisms can always be composed into a single homomorphism between syntax and semantics, the representational language is (in principle) eliminable and as Henk Zeevat (1989, p. 95) says:

> The level of representation has thereby only a secondary status in the theory: it helps to develop the grammar by making it more perspicuous, but one can in principle eliminate it in favour of the homomorphism from syntax to semantics it induces.

It is this possibility (through the use of particular compositional translations) of providing a direct homomorphism from syntax to semantics that has been held as the distinctive sign that one has accomplished a *useful* compositional interpretation. The use of mathematical algebras (as used by Montague) for the syntax and semantics has normally been considered ancillary to this main concern.

DRT does provide translations from syntactic structures to DRSs (the DRS-construction algorithm) and from DRSs to truth values with respect to a particular model (the truth-conditional interpretation). However, these translations are not compositional in the sense described above: We can not derive a single homomorphism from syntax to semantics.

It should be emphasised here that compositionality has generally been viewed as a question that can be only directed at a particular theory not a particular problem. Theorists have frequently questioned whether a linguistic problem has a compositional theory but this has generally only been determinable by providing a particular compositional theory for that problem. For instance, Kamp (1990, p. 119) admits that:

> ...it was consciously (and reluctantly) that in 1980 I abandoned the compositionalist paradigm, because I had come to believe that certain linguistic problems...required the departure from that paradigm...

Unfortunately, later theories (for example Barwise (1987) and Zeevat (1989)) have provided compositional treatments that cover the same linguistics problems investigated by Kamp (1981).

[14] Although, there is evidence that the concepts were discussed much earlier, for example see (Matilal & Sen, 1990).

However, no such theories (as yet) have been given for the coverage of the extended DRT (1993), which allows Kamp to retain the above claim for the more complex linguistic problems he covers.

Compositionality has been advocated by theorists for several reasons. For instance, Groenendijk and Stokhof (1990b, p. 102) in discussing the advantages of compositionality, state:

> One such reason can be found in computational requirements on the semantics of discourses, or texts. For example,...one would like to to be able to interpret a text in an on-line manner, i.e., incrementally, processing and interpreting each basic unit as it came along, in the context created by the interpretation of the text so far.

Meanwhile, Zeevat (1989, p. 96) succinctly groups together a number of reasons, when he says: "one can mention here the relative ease by which properties of the grammar can be proved, extendibility, comparison with other work and incorporation of other analyses". However, the entire debate over compositional or non-compositional theories may be redundant. Zadrozny (1994) proves that any semantics can be encoded as a compositional semantics, and thus compositionality is formally vacuous. His proof follows from the fact that compositionality places no restrictions on the complexity of the interpretation function. Zadrozny then derives a stronger notion of a systematic semantics which does provide a useful constraint on the construction of the interpretation function for a semantics.

Kamp believes his representations are not just non-eliminable structures within his theory but have an important philosophical role to play. He claims that DRT's representations, in some fundamental way, correspond to "the mental representations which speakers form in response to the verbal inputs they receive"[15]. Groenendijk and Stokhof (1990b, pp. 104-105) discuss two forms that such a *mentalistic* approach might take. A *weak* mentalistic approach is that language users may well manipulate representations but that these representations are not at odds with a semantic theory providing an adequate explanation of linguistic meaning without recourse to representations. An alternative, *strong* mentalistic approach claims that representations are utilized by language users and that creating fundamentally similar representations is integral to any semantic theory which attempts to explain linguistic meaning. Groenendijk and Stokhof place themselves (if anywhere) on the side of the weak mentalistic approach[16] while Kamp places himself firmly within the strong mentalistic approach (Kamp, 1990, p. 130). Given that the difficulty of classifying whether linguistic problems are solvable by compositional theories, it seems as if the debate will only be decided by the existence or not of suitable compositional theories that solve linguistic problems which representationalist's, such as Kamp, deem unsolvable in a direct homomorphic manner.

[15]See Kamp (1981, p. 282).

[16]See Groenendijk and Stokhof (1990b, p. 105).

Chapter 4

An Overview of Graph-Theoretic Semantics

This chapter will provide an overview of a new framework of discourse anaphora which will be known as Graph-Theoretic Semantics, GTS for short[1]. The semantics will be a model-theoretic semantics. In the discussions of the last chapter it was argued that any theory of discourse anaphora, when compared to the methodology and philosophy of Montague semantics, will most likely contain non-traditional (i.e., non-Montagovian) aspects to its formalisation. The important consideration, therefore, was not that non-Montagovian aspects might exist within a theory of discourse anaphora but what form amongst many possible variations they might take. Some non-Montagovian extensions might be preferable over others, because some aspects of the Montagovian tradition can be considered more valuable than others when dealing with the semantics of discourse. For instance, two principal aspects of Montague's semantics are its compositionality and its emphasis on truth-conditionality. For the semantics of discourse, compositionality is still highly valued[2] though not always considered achievable while the limitation to only a truth-conditional viewpoint on meaning is normally considered an unworkable limitation. Both DPL and DRT have shown contrasting attacks on the problem of deriving, storing and utilizing anaphoric information.

The general structure of most model-theoretic denotational semantic theories of language can be described by the components shown in figure 4.1. The semantic theory usually consists of representational and denotational structures and translations mapping between syntactic representations and semantic representations, and between semantic representations and denotations. Figure 4.1 also illustrates where DRT, DPL and GTS place anaphoric information. DRT locates anaphoric information within the representational structures, while DPL through its use of assignment function places anaphoric information in the mapping from semantic representations to denotations. GTS, as will be described in the coming chapters, is a framework which fully locates all the required anaphoric information within the denotational structures.

This chapter will provide an overview of GTS by first looking at the particular representational and denotational structures used and then looking at the nature of the mappings between these structures. The following two chapters will then look in greater detail at the semantic

[1] A simplified sentential-only GTS framework is described in Cox (1995).

[2] DPL having been created to show that a compositional theory of anaphora was possible.

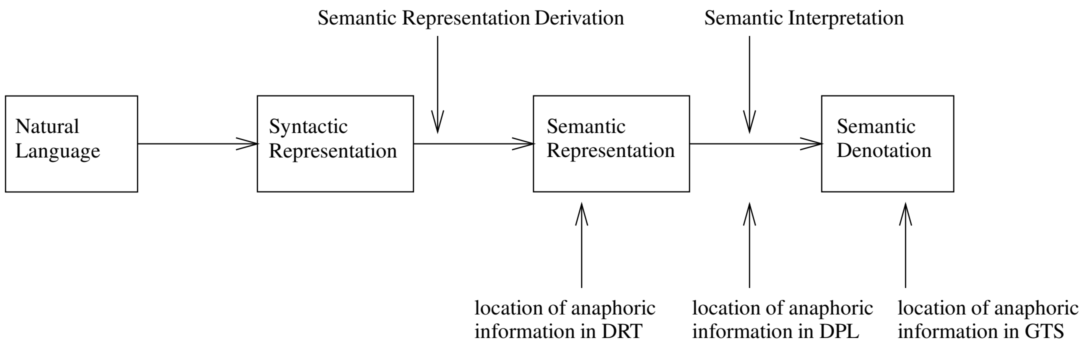


Figure 4.1: The components of a semantic theory and the location of anaphoric information within DRT, DPL and GTS.

interpretation rules which define the mapping between the semantic representations and the denotations.

4.1 The Representational Structures

The semantic representations in GTS will be given as unification feature structures. Unification feature-based representations have been used extensively for syntactic grammars (for example, LFG (Kaplan & Bresnan, 1982), PATR (Shieber *et al.*, 1983), FUG (Kay, 1985), GPSG (Gazdar *et al.*, 1985) and HPSG (Pollard & Sag, 1987)). Feature structures are standardly notated as attribute value matrices as illustrated by the following example.

(167) $$\begin{bmatrix} \mathsf{cat} & \mathsf{np} \\ \mathsf{agreement} & \begin{bmatrix} \mathsf{number} & \mathsf{NUM} \\ \mathsf{person} & \mathsf{third} \end{bmatrix} \end{bmatrix}$$

The above feature structure contains *features* such as cat and *values* such as np. Features which take atomic values such as the feature cat are called atomic features. Features which take other feature structures as their values, such as agreement, are called complex features. Features within a feature structure which are not assigned particular values can be represented as having variable values, such as NUM for the feature number. An important aspect of feature structures is that they can be *reentrant*. That is, a feature structure can contain features which are forced to take the same value. Features are marked as reentrant by specifying identical variable names for the particular features concerned.[3] For a more in-depth introduction to unification feature structures as well as the operations which can be defined over them the reader is prompted to see Shieber (1986). Their use for semantic purposes can be found in Shieber (1986), Pollard and Sag (1987) and Fenstad et al. (1987).

Nerbonne (1992) discusses certain advantages of having a feature-based syntax/semantics interface, which can be summarized as

- Feature structures allow a flexible and efficient specification of constraints.

[3] It should be noted that the descriptional conventions taken here to describe features structures are but one of many possibilities.

- A feature-based semantic representation harmonizes well with current day feature-based syntactic descriptions.
- Meanings can be underspecified in a manner difficult to achieve with other semantic representations.

In GTS, the semantic information for a constituent is held within a complex feature sem. For instance, a possible sem feature built after the analysis of the sentence in (168) is given in (169).

(168) Every farmer owns a donkey.

(169)

$$
\text{sem}\;\left[\begin{array}{ll}
\text{control} & \left[\begin{array}{ll}
\text{subject} & \left[\begin{array}{ll}\text{reading} & \text{distributive}\\ \text{pol} & +\\ \text{uniq} & -\end{array}\right]\\
\text{object} & \left[\begin{array}{ll}\text{reading} & \text{distributive}\\ \text{pol} & +\\ \text{uniq} & -\end{array}\right]\\
\text{predicate} & \left[\begin{array}{ll}\text{pred} & \text{own}\\ \text{pol} & +\\ \text{scope} & \text{subjectwide}\end{array}\right]
\end{array}\right]\\
\text{arg1} & \left[\begin{array}{ll}
\text{control} & \left[\begin{array}{ll}\text{pred} & \text{every}\\ \text{uniq} & -\\ \text{pol} & +\end{array}\right]\\
\text{arg1} & \left[\begin{array}{ll}\text{control} & \left[\begin{array}{ll}\text{pred} & \text{farmer}\end{array}\right]\end{array}\right]_{n}
\end{array}\right]_{det}\\
\text{arg2} & \left[\begin{array}{ll}
\text{control} & \left[\begin{array}{ll}\text{pred} & \text{a}\\ \text{uniq} & -\\ \text{pol} & +\end{array}\right]\\
\text{arg1} & \left[\begin{array}{ll}\text{control} & \left[\begin{array}{ll}\text{pred} & \text{donkey}\end{array}\right]\end{array}\right]_{n}
\end{array}\right]_{det}
\end{array}\right]_{tv}
$$

The semantic representation is built up in a strict compositional manner in tandem with the syntactic analysis, utilizing basic *typed* feature structures of the form:

$$
\left[\begin{array}{l}\text{control}\\ \text{arg1}\\ .\\ .\\ .\\ \text{argN}\end{array}\right]_{type}
$$

The *type* of the feature structure is given in the bottom right-hand corner of the matrix. Some example types are tv for transitive verb, det for determiner and n for (lexical) noun. The control feature holds certain ancillary features which determine the particular interpretation given to the predicate in which the particular control feature is contained. Following this there are zero or more numbered arg features which provide the semantic representations of the arguments to the predicate. The particular features that can appear within the control complex feature will not be discussed in detail here. However, for the particular case of transitive verbal predicates, the control complex feature is split between subject, object and predicate features which contain features such as the following:

- reading - the verbal reading.
- pol - the polarity.
- scope - the quantifier scope.
- uniq - whether there must be a uniqueness constraint.

The feature-based representation structure has been constructed to solely fulfill the needs of the anaphoric semantics developed i.e., to allow a compositional interpretation and to allow the effective stipulation of the representational constraints needed to interpret different linguistic constituents in simple anaphoric discourse. How this structure is built and utilized will be discussed further in sections 4.3 and 4.4.

4.2 The Denotational Structures

The framework proposes to maximise the use of denotational structures in describing the anaphoric information derived from the interpretation of a discourse. Previous model-theoretic semantics devoted to discourse and anaphora have tended to minimise the complexity of the denotation structures. In the case of DRT this has led to the retainment of the representational structures, as only here is there the required constraint information connecting the different denotational structures produced by any interpretation. DPL takes a step towards the denotational route but stops half-way by centering the anaphoric information within assignment functions which provide a link between representations and denotations. The aim here is that the denotations themselves should provide all the required information for anaphoric analysis.

First though, one has to decide what sort of denotational information is needed for discourse anaphora. If we look at the simple example below:

(170) Jack owns a donkey. Jane owns a horse. They beat them.

To handle the anaphors in the third sentence in (170) we need to identify individuals for *Jack*, *Jane*, *a donkey* and *a horse*. For a referential reading of the third sentence this is all we need (plus a way of combining these individuals into collections). However, for a bound anaphoric reading of the third sentence we also need to know the relationships between the individuals identified. This means that the denotational space needs not only to identify sets of individuals but also the relationships between them.

Given these requirements, a basic denotational description of anaphoric information seems clear: sets of individuals and relationships between them. However, isn't this information already provided within the model? A standard set-theoretic model structure is $\langle D, F\rangle$, where D is a set of individuals and F provides some basic semantic denotation for the non-logical predicates of the semantic representation language. For binary predicates, F might provide a set of pairs of individuals in D, thus specifying which individuals satisfy a particular predicate. Unfortunately, models are complete descriptions of a world or domain unlike discourses, a point emphasised by the truth-conditional interpretation of DRT being described as an *embedding* of a discourse inside a model. That is, anaphoric information can be viewed as partial information derived by the interpretation of a discourse.

Furthermore, some mechanism is needed to individuate the particular antecedents. That is, anaphoric information is bundled together into what are called antecedents. In DRT, the antecedents are identified by particular discourse referents, while in DPL they are identified by particular variables. In representational approaches variables (or discourse referents) provide the required individuation. They provide distinct flags within the representation, each derived variable describing a possible anaphoric antecedent. We are seeking an equivalent denotational treatment of antecedent individuation. However, models do not provide any means of identifying particular antecedents.

Finally, a model is a static structure. We need some structure which can handle the dynamic aspect of anaphoric information manipulation within a discourse, i.e. the fact that the information available is changing from sentence to sentence during the interpretation of a discourse. To summarise, there are three reasons why models do not fit the requirements of a denotational anaphoric information provider.

1. Anaphoric denotations are partial-models.
2. Anaphoric denotations are individuated as antecedents.
3. Anaphoric analysis has dynamic aspects.

In the following sections, I will first begin by discussing the particular model structure I shall be utilizing before discussing the particular denotation structures utilized by the framework. In section 4.2.2, I will define a denotational structure called a **denotation graph** which will cover the first two requirements along with a denotational structure called a **discourse space** which in combination with the semantic interpretation will cover the third requirement.

4.2.1 Model Structure

Within this section, I will discuss the particular type of model structure I will be using with respect to which the model-theoretic semantics will be defined.

Definition 2: A model is a structure $\langle D, F\rangle$, where D is a set of individuals and F is a function which assigns a *basic semantic value* to the (non-logical) predicates utilized by the semantics.

It was shown in section 2.3.1 of chapter 2 that a set-theoretic model domain, D, is adequate for the analysis of plural count nouns. A set-theoretic domain is utilized here. The function F will assign a basic semantic value to the (non-logical) predicates of the semantic representation language. Within the present work lexical nouns will be treated as zero-place predicates, i.e., predicates that take zero arguments. For zero-place predicates, such as the nominal predicate farmer, F will assign a set of individuals. For one-place predicates, such as the unary predicate gather, F will assign a set of sets of individuals. For two-place predicates, such as the binary predicate beat, F assigns a set of pairs of sets (of individuals) from the domain, D. As an example, a possible model M_1 is given below.

(171) $M_1 = \langle D_1, F_1 \rangle$, where
$D_1 = \{a, b, c, d, e, f\}$,
$F_1(\mathsf{farmer}) = \{a, b, c\}$
$F_1(\mathsf{donkey}) = \{d, e, f\}$
$F_1(\mathsf{gather}) = \{\{d, e, f\}, \{a\}\}$
$F_1(\mathsf{sleep}) = \{\{a\}, \{b\}\}$
$F_1(\mathsf{own}) = \{\langle\{a\}, \{d\}\rangle, \langle\{b\}, \{e\}\rangle, \langle\{c\}, \{f\}\rangle, \langle\{a, b\}, \{d, e\}\rangle\}$
$F_1(\mathsf{beat}) = \{\langle\{a\}, \{d\}\rangle, \langle\{b\}, \{e\}\rangle, \langle\{c\}, \{f\}\rangle, \langle\{a, b\}, \{d, e\}\rangle\}$

Within the semantic interpretation to be derived later, if we have a set, say $\{a, b\}$, and wish to check distributively whether these individuals sleep, then utilizing the above model we will check whether the singleton sets $\{a\}$ and $\{b\}$ are members of $F(\mathsf{sleep})$. If we wished to check collectively that $\{a, b\}$ sleep we would check whether the set $\{a, b\}$ is a member of $F(\mathsf{sleep})$. In this manner, distributive and collective readings are treated in a uniform manner. This uniformity is facilitated by the model structure in which nominal predicates, such as farmer, are zero-place predicates while intransitive verbal predicates, such as sleep, are one-place predicates. This separation is further encouraged by the semantic representation language which does not contain any representational structures carrying out the traditional role of variables. For example, in predicate calculus the predicates derived for the lexical noun farmer and the intransitive verb sleep are traditionally both one-place predicates over a single variable. While in GTS, the semantic representation for the lexical noun farmer is a predicate with no arg features. The semantic representation for the intransitive verb sleep, however, would be a predicate with a single arg feature containing the semantic representation of the single argument to the verbal predicate.

4.2.2 Denotation Graphs and Discourse Spaces

Given that the model does not directly provide an adequate means of specifying anaphoric information derived from the interpretation of a discourse and we wish in GTS to place all the required anaphoric information within the denotations, an appropriate denotational structure is required. An obvious candidate, given that we need sets of individuals and relationships between them, is a network or graph based structure. The new denotational structure will be called a **denotation graph**[4]. Each vertex will be described by a particular identifier and a set of sets of individuals from a model. Identifiers serve the purpose of distinguishing between vertices containing identical sets of individuals. I will assume the semantic interpretation to be discussed in detail in the next chapter is carried out with respect to a model and a set of identifiers. Before giving the definition of a vertex the notion of a *denotation set* is provided. A denotation set will describe the individuals from a model that are contained within a vertex. The definition is given below with respect to a model $M = \langle D, F \rangle$.

Definition 3: A *denotation set* is a set of sets of individuals from D.

The definition of a vertex can now be given with respect to a set of identifiers I.

Definition 4: A vertex v is a tuple $\langle i, \mathcal{C} \rangle$ where i is an identifier and $\mathcal{C}$ is a denotation set.

[4]In unambiguous circumstances I shall simply use the term *graph* in place of *denotation graph*.

The edges connecting vertices will be of two types: relational edges and anaphoric edges. Relational edges will describe some (binary) relationship between the sets of individuals described by the vertices the edge connects.

Definition 5: A relational edge is a triple $\langle v, v', R\rangle$, where v and v' are vertices and R is a binary relation over pairs of sets of individuals in D (the model domain).

Anaphoric edges determine which vertices are anaphorically dependent on each other. An anaphoric edge is defined below.

Definition 6: An anaphoric edge is a pair $\langle v, v'\rangle$ where v and v' are vertices.

It will always be the case that the first member of the pair in an anaphoric edge (v above) will be the vertex describing some anaphor in the discourse being interpreted and the second member of the pair in the anaphoric edge (v' above) will be the vertex describing some antecedent in the discourse being interpreted. A denotation graph can now be defined.

Definition 7: A denotation graph G will be a triple $\langle V, E, A\rangle$, where V is a set of vertices, E is a set of relational edges and A is a set of anaphoric edges.

Some examples will help to make concrete the possible denotation graphs that the semantic interpretation of particular linguistic objects might derive. Given that denotation graphs are model-dependent structures, I will utilize the previously given model, M_1, shown again in (172) and a set of identifiers I_1 given in (173).

(172) $M_1 = \langle D_1, F_1\rangle$, where
$D_1 = \{a, b, c, d, e, f\}$,
$F_1(\mathsf{farmer}) = \{a, b, c\}$
$F_1(\mathsf{donkey}) = \{d, e, f\}$
$F_1(\mathsf{gather}) = \{\{d, e, f\}, \{a\}\}$
$F_1(\mathsf{sleep}) = \{\{a\}, \{b\}\}$
$F_1(\mathsf{own}) = \{\langle\{a\}, \{d\}\rangle, \langle\{b\}, \{e\}\rangle, \langle\{c\}, \{f\}\rangle, \langle\{a, b\}, \{d, e\}\rangle\}$
$F_1(\mathsf{beat}) = \{\langle\{a\}, \{d\}\rangle, \langle\{b\}, \{e\}\rangle, \langle\{c\}, \{f\}\rangle, \langle\{a, b\}, \{d, e\}\rangle\}$

(173) $I_1 = \{1, 2, 3\}$

With respect to M_1 and I_1, the interpretation of the phrases *every farmer* and *a donkey* will derive denotation graphs G_1 and G_2 given below.

(174) $G_1 = \langle\{\langle 1, \{\{a, b, c\}\}\rangle\}, \{\}, \{\}\rangle$

(175) $G_2 = \langle\{\langle 2, \{\{d\}, \{e\}, \{f\}\}\rangle\}, \{\}, \{\}\rangle$

These phrases derive single vertex graphs whose single vertices are derived from the model by appropriate interpretation rules, to be given later. The graphs G_1 and G_2 can also be displayed diagrammatically as shown in figure 4.2. The vertices of a denotation graph, each of which describes a denotation set are displayed as circles with their identifiers given along side. The contents of the vertices are the sets within the denotation set. The graphs are also named to help readability and enclosed in a box. A sentence such as (176) below will derive, under an appropriate interpretation, a denotation graph G_3 as shown in (177), with respect to M_1.

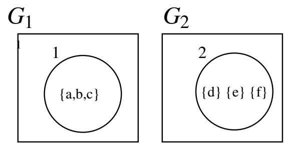


Figure 4.2: Denotation graphs for *every farmer* (G_1) and *a donkey* (G_2) with respect to M_1 and I_1.

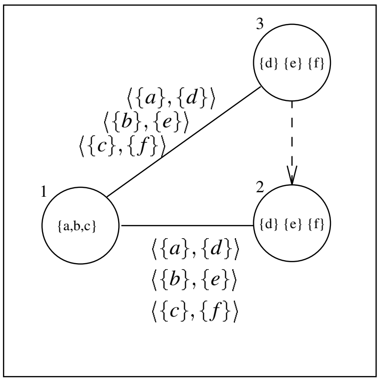


Figure 4.3: Denotation graph for *every farmer who owns a donkey beats it* with respect to M_1 and I_1.

(176) Every farmer who owns a donkey beats it.

(177) $G_3 = \langle\{\langle 1,\{\{a,b,c\}\}\rangle,\langle 2,\{\{d\},\{e\},\{f\}\}\rangle,\langle 3,\{\{d\},\{e\},\{f\}\}\rangle\},$
$\{\langle\langle 1,\{\{a,b,c\}\}\rangle,\langle 2,\{\{d\},\{e\},\{f\}\}\rangle,\{\langle\{a\},\{d\}\rangle,\langle\{b\},\{e\}\rangle,\langle\{c\},\{f\}\rangle\}\rangle,$
$\langle\langle 1,\{\{a,b,c\}\}\rangle,\langle 3,\{\{d\},\{e\},\{f\}\}\rangle,\{\langle\{a\},\{d\}\rangle,\langle\{b\},\{e\}\rangle,\langle\{c\},\{f\}\rangle\}\rangle\},$
$\{\langle\langle 3,\{\{d\},\{e\},\{f\}\}\rangle,\langle 2,\{\{d\},\{e\},\{f\}\}\rangle\rangle\}\rangle$

The graph in (177) can be displayed diagrammatically as shown in figure 4.3. Relational edges show the particular relation defined between the two vertices. Although relational edges are directional in nature I will omit marking their direction: the diagrams will always be given in such a way that it will be obvious which set from each pair is derived from which vertex. Anaphoric edges are marked as arrowed dashed lines between the appropriate vertices. The direction of the line indicated by the arrow is from anaphor to antecedent. If only a single graph is being displayed a graph name will not be given.

Sometimes a graph will be displayed not with respect to a particular model but with respect to the abstract notion of a satisfying model for a particular interpretation. This allows the graphs derived from particular discourses to be discussed without reference to a particular model. A satisfying model for a sentence is any model in which the semantic interpretation derives a denotation graph containing no empty vertices. The sentence in (176) will derive in a satisfying model a graph which can be displayed as in figure 4.4. The relational edges are marked with the semantic predicate from which they have been derived. The vertices are marked with vertex labels instead of particular identifiers. As the figure illustrates the graph derived for (176) within any satisfying model, no model specific information is provided.

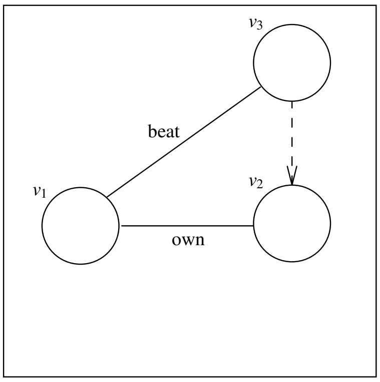


Figure 4.4: Denotation graph for *every farmer who owns a donkey beats it* with respect to a satisfying model.

Denotation graphs will be derived from the semantic interpretation of a discourse. As the denotation graphs hold all the anaphoric information, a place to store all these various graphs is required. This place I will call a **discourse space**. Formally, a discourse space will simply be **a set of denotation graphs**. The discourse space existing at any point in the analysis of a discourse will hold the anaphoric information derived from the interpretation of that discourse up to this point.

We can now review the three requirements for denotational anaphoric information given in section 4.2, repeated below.

1. Anaphoric denotations are partial.

2. Anaphoric denotations are individuated as antecedents.

3. Anaphoric analysis has dynamic aspects.

Denotation graphs certainly only provide part of the information described in the model itself and thus satisfy the first requirement. To satisfy the second requirement, we will have to decide what denotational structures antecedents will be associated with. The decision taken is that each vertex in each denotation graph in a discourse space can be seen as a possible antecedent for an anaphor. The discourse space partially satisfies the third requirement. The semantic interpretation will define exactly how the discourse space captures the dynamic change in anaphoric information through the analysis of a discourse.

4.3 Constructing the Semantic Representation

This section will outline how the semantic representations can be constructed during a syntactic analysis. The feature-based nature of the semantic representations allows a variety of possible ways in which the semantic representations could be included within a feature-based syntactic grammar. I will describe only one possible solution. The GTS framework should not be considered as advocating this particular example integration as the preferred method. The example provided is meant to highlight the simplicity of integrating the construction of the semantic representations into a syntactic analysis in a compositional manner. It is not supposed to be an advocation of that compositional solution over any other.

I have chosen to utilize the PATR unification grammar formalism (Shieber *et al.*, 1983; Shieber, 1986) for the purpose of providing a compositional analysis. I will begin by presenting here PATR lexical rules which describe nominal, verbal and determiner predicates. The general structure of the feature matrix for a linguistic constituent is given below.

(178)
$$\left[\begin{array}{ll} \textsf{syn} & \textsf{SYNTAX} \\ \textsf{sem} & \textsf{SEMANTICS} \\ \textsf{subcat} & \textsf{SUBCAT} \end{array}\right]$$

The syn feature contains appropriate syntactic information, while the sem feature contains the semantic information. The subcat feature contains subcategorization information which is utilized by the PATR grammar as outlined below. The appropriate lexical rules for some example predicates of each type are shown below the predicate concerned. Appropriate basic syntactic values are included in the definitions as well.

(179)
$$\left[\begin{array}{ll} \textsf{control} & \left[\begin{array}{ll} \textsf{pred} & \textsf{farmer} \\ \textsf{number} & \textsf{singular} \end{array}\right] \end{array}\right]_{n}$$

```
Word farmer: <cat> = n
             <head sem type> = n
             <head sem control pred> = farmer
             <head sem control number> = singular
             <head syn number> = singular
             <head syn person> = third.
```

(180)
$$\left[\begin{array}{ll} \textsf{control} & \left[\begin{array}{ll} \textsf{pred} & \textsf{every} \\ \textsf{uniq} & - \\ \textsf{pol} & + \\ \textsf{read} & \textsf{distributive} \\ \textsf{negreading} & \textsf{NR} \end{array}\right] \\ \textsf{arg1} & \textsf{ARG1} \end{array}\right]_{det}$$

```
Word every: <cat> = det
        <head sem type> = det
        <head sem control pred> = every
        <head sem control uniq> = no
        <head sem control pol> = positive
        <head sem control reading> = distributive
        <head sem arg1> = <subcat first head sem>
        <head syn number> = singular
        <subcat first cat> = nbar
        <subcat rest> = end.
```

(181)
$$\left[\begin{array}{ll}
\mathsf{control} & \left[\begin{array}{ll}
\mathsf{subject} & \left[\begin{array}{ll}
\mathsf{reading} & \mathsf{SREADS} \\
\mathsf{pol} & \mathsf{POLS} \\
\mathsf{negreading} & \mathsf{NRS} \\
\mathsf{uniq} & \mathsf{USS}
\end{array}\right] \\
\mathsf{object} & \left[\begin{array}{ll}
\mathsf{reading} & \mathsf{SREADO} \\
\mathsf{pol} & \mathsf{POLO} \\
\mathsf{negreading} & \mathsf{NRO} \\
\mathsf{uniq} & \mathsf{USO}
\end{array}\right] \\
\mathsf{predicate} & \left[\begin{array}{ll}
\mathsf{pred} & \mathsf{owns} \\
\mathsf{scope} & \mathsf{SCOPE} \\
\mathsf{pol} & + \\
\mathsf{negreading} & \mathsf{NRV} \\
\mathsf{aarel} & \mathsf{AAREL}
\end{array}\right]
\end{array}\right] \\
\mathsf{arg1} & \mathsf{ARG1} \\
\mathsf{arg2} & \mathsf{ARG2}
\end{array}\right]_{tv}$$

```
Word owns: <cat> = v
        <head syn form> = finite
        <head syn number> = singular
        <head sem type> = tv
        <head sem control pol> = positive
        <head sem control pred> = own
        <head sem arg1> = <subcat rest first head sem>
        <head sem arg2> = <subcat first head sem>
        <subcat rest first head syn number> = singular
        <subcat rest first head syn person> = third
        <subcat rest rest> = end.
```

There are several differences between the semantic representations described as feature matrices and the semantic representations in PATR. I have restricted my PATR structures to those understandable by the Miniature PATR-II system which is used by the computational implementation of the framework discussed in chapter 7. To this end, I have replaced $+$ with `yes` and $-$ with `no` in all features that require a binary value except the pol feature which has been given `positive` and `negative` values, which relate more closely to the meaning of this feature. Furthermore, the type of a feature structure is given by a feature `type`. I have also assumed for the present exposition that many features within the control structure of a verbal predicate receive their values from the control features of their arguments. For instance, the reading (verbal reading), uniq (uniqueness constraint) and pol (polarity) features of the subject and object arguments receive their values from the values given by their respective subject and object arguments. These unification constraints have been integrated into the PATR grammar rules given below which provide a very basic coverage of some simple English syntax.

```
RULE {sentence matrix}
 S -> NP VP:
```

```
    <S head> = <VP head>
    <S head syn form> = finite
    <VP subcat first> = <NP>
    <VP subcat rest> = end
    <S head sem control subject> = <NP head sem control>
    <NP head syn rel> = false.

 RULE {sentence relative}
  S -> NP VP:
    <S head> = <VP head>
    <S head syn form> = finite
    <S subcat> = <VP subcat>
    <NP head syn rel> = true.

 Rule {transitive verb phrase}

 VP_1 -> V NP:
    <VP_1 head> = <V head>
    <V subcat first> = <NP>
    <VP_1 subcat> = <V subcat rest>
    <VP_1 head sem control object> = <NP head sem control>.

 Rule {Negative verb}

 V_3 -> V_1 Neg V_2:
    <V_1 head form> = aux
    <V_2 head syn form> = base
    <V_3 head sem > = <V_2 head sem>
    <V_3 subcat> = <V_2 subcat>
    <V_3 subcat rest first head syn number> = <V_1 head syn number>
    <V_3 head sem control predicate pol> = negative.

 Rule {Noun phrase}

 NP -> Det Nbar:
  <NP head> = <Det head>
  <Det head syn number> = <Nbar head syn number>
  <NP head sem control number> = <Nbar head syn number>
  <NP head syn rel> = false
  <Det subcat first> = <Nbar>
  <Det subcat rest> = end.

Rule {Proper Noun}

NP -> PN:
 <NP head> = <PN head>
 <NP head syn rel> = false.

 Rule {Nbar lexical noun}
```

```
 Nbar -> N:
  <Nbar head> = <N head>.

 Rule {Relative clause combination}

 Nbar_1 -> Nbar_2 S:
   <Nbar_1 head> = <S head>
   <S subcat first> = <Nbar_2>
   <S subcat rest> = end.
```

For completeness, the relative pronoun *who*, the auxiliary verbs *do* and *does* and the negative *not* are shown below under their PATR definitions. Note, none of these words introduce semantic predicates.

```
Word who: <cat> = np
    <head syn rel> = true.

Word does: <cat> = v
 <head syn form> = aux
 <head syn number> = singular.

Word do: <cat> = v
 <head syn form> = aux
 <head syn number> = plural.

Word not: <cat> = neg.
```

The grammar can be found in full in appendix B.

4.4 The Semantic Interpretation

Within, the following sections I shall describe the semantic interpretation of the semantic representations. The linguistic constituents and data covered by these representations are restricted to those covering noun phrases, simple relative clauses, and transitive verbs. The analysis of transitive verbs will cover amongst other things simple forms of negation, various readings (e.g., distributive, collective) and quantifier scopings.

I will assume a discourse is a temporally ordered set of (declarative extensional) sentences $S_1, S_2, \ldots, S_n$. An appropriate (compositional) analysis will, given a sentence in the discourse S_j, derive an appropriate feature matrix A_j representing the syntactic and semantic information of S_j. The value of the sem feature of A_j will be α_j, which will contain the semantic representation derived from S_j. That is, the analysis of a discourse $S_1, S_2, \ldots, S_n$ will produce a series of semantic representations $\alpha_1, \alpha_2, \ldots, \alpha_n$. The semantic interpretation function will be applied under the temporal ordering to each semantic representation α_j. The interpretation of each semantic representation for each sentence will derive a denotation graph which will be added to the discourse space. At the start of the discourse the discourse space is empty. It is incrementally extended through the interpretation of each sentence in the discourse. Furthermore, the interpretation of each semantic representation will be carried out in a compositional manner, each semantic predicate deriving a denotation graph which will be added to the discourse space. That is, each

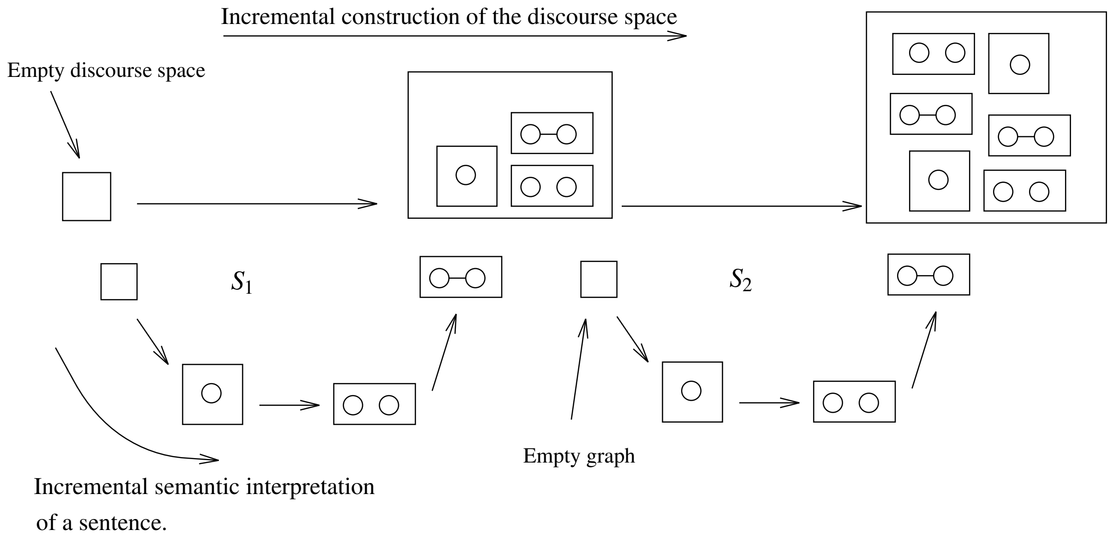


Figure 4.5: The incremental interpretation of a discourse

sentence does not derive a single denotation graph but a whole series of denotation graphs, one for each semantic predicate within the semantic predicate describing the sentence. The semantic interpretation of each sentence is begun with an empty denotation graph which is incrementally extended through the interpretation to derive the final graph describing the sentence. That is, each of the graphs derived from the interpretation of a sentence is an incremental extension of some previous one, except for the first graph which, as has been said, is empty. The process is illustrated schematically in figure 4.5. The figure illustrates the incremental construction of the discourse space through the incremental compositional interpretation of two sentences, S_1 and S_2.

4.4.1 Quantification

The quantificational analysis I shall be following will be based on the generalized quantifier framework which was primarily initiated in linguistics by Barwise and Cooper (1981) along with Higginbotham and May (1981) and Keenan and Stavi (1986). The basic theory of generalized quantifiers was discussed briefly in the previous chapter in section 3.2.4. However, to reiterate, a generalized quantifier is a set of sets of individuals (i.e., the same model-theoretic type as a denotation set in GTS). Generalized quantifiers are applied to the semantic analysis of noun phrases. Some standard generalized quantifier interpretations are as follows, given that D is the set of individuals defined by a model.

(182) $[\![\text{Every farmer}]\!] = \{X \subseteq D | X \text{ contains every farmer }\}$

(183) $[\![\text{Most farmers}]\!] = \{X \subseteq D | X \text{ contains most farmers }\}$

(184) $[\![\text{No farmers}]\!] = \{X \subseteq D | X \text{ contains no farmers }\}$

One possible way in which to utilize generalized quantifiers is given by Barwise and Cooper, and can be illustrated by the following simple example.

(185) Every farmer runs.

Barwise and Cooper require that verb phrases and nominal phrases each denote a set of individuals. That is, in (185) the verb phrase *runs* denotes the set of individuals (from a particular model) that run, while the lexical noun *farmer* denotes the set of farmers in the model domain, D. In determining the truth or falsity of the whole sentence we check if the set of individuals associated with the verb phrase *runs* is a member of the set of sets denoted by the generalized quantifier *every farmer*. Therefore, when discussing generalized quantifiers, it is customary to use the notation $Det_D AB$ to mean a determiner *Det* over domain D applied to sets A and B. In other words, B is a member of the generalized quantifier determined by $Det_D A$.

Barwise and Cooper proposed a series of universal constraints on generalized quantifiers, and thus in consequence on the semantics of noun phrases of natural languages. One of the most important is the "lives on" property (Barwise & Cooper, 1981, p. 178), also known as the Conservativity universal (Keenan & Stavi, 1986, p. 276).

Definition 8: *Conservativity Universal*

$$Det_D AB \Leftrightarrow Det_D A(A \cap B)$$

This definition states that the quantifier $Det_D A$ applied to the set B is equivalent to the quantifier $Det_D A$ applied to the set A intersected with the set B. This universal hypothesis (if correct) ensures that in manipulating generalised quantifiers we can concern ourselves solely with the individuals denoted by the lexical nouns (or proper names). Looking back at (185), this means that in determining the denotation of the verb phrase *runs* we need only consider those individuals that run and are farmers. In evaluating a more complex example such as *most farmers own a donkey* we need only concern ourselves with those individuals from the model which are farmers or donkeys.

Some examples of the semantic representation given to noun phrases in GTS are shown below.

(186) Every farmer

$$\left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll}\text{pred} & \text{every}\\ \text{uniq} & -\\ \text{pol} & +\end{array}\right]\\ \text{arg1} & \left[\text{control}\ \left[\text{pred}\ \ \text{farmer}\right]\right]_{n} \end{array}\right]_{det}$$

(187) A farmer

$$\left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll}\text{pred} & \text{a}\\ \text{uniq} & -\\ \text{pol} & +\end{array}\right]\\ \text{arg1} & \left[\text{control}\ \left[\text{pred}\ \ \text{farmer}\right]\right]_{n} \end{array}\right]_{det}$$

(188) No farmers

$$\left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll}\text{pred} & \text{no}\\ \text{uniq} & -\\ \text{pol} & -\end{array}\right]\\ \text{arg1} & \left[\text{control}\ \left[\text{pred}\ \ \text{farmer}\right]\right]_{n} \end{array}\right]_{det}$$

(189) three farmers

$$\begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{three} \\ \textsf{uniq} & \textsf{U} \\ \textsf{pol} & + \end{bmatrix} \\ \textsf{arg1} & \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{farmer} \end{bmatrix} \end{bmatrix}_{n} \end{bmatrix}_{det}$$

As can be observed, each quantifier representation has a control complex feature with three features, pred, uniq and pol. The pred feature contains the particular determiner predicate, e.g., every, or no. The particular importance of the uniq and pol features will be explained below. The noun phrase *three farmers* has two possible representations corresponding to reading the determiner *three* to be either *exactly three* or *at least three*, respectively. For the former reading the feature uniq has the value $+$, while for the latter the feature uniq has the value $-$. However both representations of the noun phrase *three farmers* will produce the same denotation, the difference between the two readings being handled at the level of verbal relations.

In handling the denotation of the above noun phrases I shall be using the concept of a *witness set* as defined by Barwise and Cooper (1981, p. 191).

(190) A *witness set* for a quantifier $Det_D A$ living on A is any subset w of A such that $w \in Det_D A$.

The manner in which I shall utilize witness sets will depend on a classification of quantifiers given by Barwise and Cooper termed monotonicity (Barwise & Cooper, 1981, p. 184-191). Quantifiers can be classified as to whether they are monotone increasing, monotone decreasing or non-monotone.

- A quantifier Q is *monotone increasing* if $X \in Q$ and $X \subseteq Y \subseteq D$ implies $Y \in Q$.
- A quantifier Q is *monotone decreasing* if $X \in Q$ and $Y \subseteq X \subseteq D$ implies $Y \in Q$.
- A quantifier A is non-monotone if it is neither monotone increasing nor monotone decreasing.

When interpreting a noun phrase in GTS, a denotation set $\mathcal{C}$ is derived, where $\mathcal{C}$ is determined as below for particular quantifiers.

(191) If $Det_D A$ is monotone increasing or non-monotone then $\mathcal{C} = \{w | w \in Det_D A\}$

(192) If $Det_D A$ is monotone decreasing then $\mathcal{C} = \{A - w | w \in Det_D A\}$

For monotone increasing and non-monotone quantifiers GTS collects together the witness sets applicable to the quantifier. However, for monotone decreasing quantifiers GTS collects the complements of the witness sets with respect to the set A. In the subsequent discussion unless explicitly mentioned I shall use witness set to mean those sets that GTS derives for the denotation of quantifiers, not those derived by Barwise and Cooper, which differ with respect to the monotone decreasing quantifiers.

Denotation sets derived within GTS for the above noun phrases in (186) to (189) are given below, where $[\![farmer]\!]$ is to be interpreted here as the set of farmers in a particular model.

(193) every farmer : $\{[\![farmer]\!]\}$

(194) No farmers: $\{[\![farmer]\!]\}$

(195) a farmer : $\{\{x\}|x \in [\![farmer]\!]\}$

(196) three farmers: $\{X \subseteq [\![farmer]\!]||X| = 3\}$

There are two things to note from the above denotations. The denotations for *every farmer* and *no farmer* are the same and there is only a single denotation for *three farmers* even though there are two possible readings for *three farmers*, i.e., *exactly three farmers* or *at least three farmers*[5] To understand how the required readings are provided it must be understood how the denotations will be utilized by any verbal relation to which they are arguments. The witness sets contained within these denotation sets will be utilized during the analysis of verbal relations in determining those sets that satisfy a verbal reading. This will be discussed further in the next section. However, the purpose of the features pol and uniq is to pass information up to the verbal predicate which will help determine the possible interpretation required. In particular, the feature uniq specifies whether only one witness set must satisfy the verbal predicate or any number of witness sets. The feature pol helps determine the final polarity of the verbal relation. The following two examples will informally demonstrate how the required readings are derived from the denotations and the information in the semantic representations of noun phrases.

(197) (Exactly) three farmers run.

(198) No farmers run.

The semantic representation of *(exactly) three farmers* will contain the features uniq + and pol +. This requires the verbal relation to ensure that only one witness set of three farmers satisfies the predicate run and, secondly, the noun phrase contributes positively to the polarity of the verbal relation. That is, in interpreting (197) we must ensure that only a single set of three farmers satisfies the run verbal predicate. The semantic representation of *no farmers*, given in (188), contains the features uniq $-$ and pol $-$. This stipulates no uniqueness constraint is to be applied to the witness sets of this quantifier and furthermore the noun phrase contributes negatively to the polarity of the verbal relation. That is, in interpreting (198) we must essentially check that **every** farmer (see (194)) does **not** run.

4.4.2 Verbal Relations

A simple sentence is given in (199) along with one of its possible associated semantic representations in (200).

(199) Every farmer owns a donkey.

[5] I am assuming a semantic treatment of the difference between the two readings of numeral determiners such as *three*, although a common strategy is to provide a pragmatic treatment for this distinction.

(200)

$$
\begin{array}{ll}
\mathsf{sem} & \left[\begin{array}{ll}
\mathsf{control} & \left[\begin{array}{ll}
\mathsf{subject} & \left[\begin{array}{ll}\mathsf{reading} & \mathsf{distributive}\\ \mathsf{pol} & +\\ \mathsf{uniq} & -\end{array}\right]\\
\mathsf{object} & \left[\begin{array}{ll}\mathsf{reading} & \mathsf{distributive}\\ \mathsf{pol} & +\\ \mathsf{uniq} & -\end{array}\right]\\
\mathsf{predicate} & \left[\begin{array}{ll}\mathsf{pred} & \mathsf{own}\\ \mathsf{pol} & +\\ \mathsf{scope} & \mathsf{subjectwide}\end{array}\right]
\end{array}\right]\\
\mathsf{arg1} & \left[\begin{array}{ll}
\mathsf{control} & \left[\begin{array}{ll}\mathsf{pred} & \mathsf{every}\\ \mathsf{uniq} & -\\ \mathsf{pol} & +\end{array}\right]\\
\mathsf{arg1} & \left[\begin{array}{ll}\mathsf{control} & \left[\begin{array}{ll}\mathsf{pred} & \mathsf{farmer}\end{array}\right]\end{array}\right]_{n}
\end{array}\right]_{det}\\
\mathsf{arg2} & \left[\begin{array}{ll}
\mathsf{control} & \left[\begin{array}{ll}\mathsf{pred} & \mathsf{a}\\ \mathsf{uniq} & -\\ \mathsf{pol} & +\end{array}\right]\\
\mathsf{arg1} & \left[\begin{array}{ll}\mathsf{control} & \left[\begin{array}{ll}\mathsf{pred} & \mathsf{donkey}\end{array}\right]\end{array}\right]_{n}
\end{array}\right]_{det}
\end{array}\right]_{tv}
\end{array}
$$

The verbal predicate own derived from the verb *owns* has two argument representations held in arg1 and arg2 which describe the two noun phrases *every farmer* and *a donkey*. These two noun phrases are interpreted as generalized quantifiers as explained in the previous section. In GTS, they derive vertices containing denotation sets holding the appropriate witness sets for each quantifier. The analysis of a verbal predicate is to decide which witness sets from each argument generalized quantifier satisfy some particular verbal reading. The particular reading provided is determined by the features appearing within the control feature. The control features shown in (200) cover the basic possibilities. The control feature is split into subject, object and predicate complex features which concern the subject and object arguments and the verbal predicate itself. In an intuitive sense, the most prominent feature is reading which specifies the particular type of verbal reading to be given, for example distributive or collective. The pol feature determines the polarity, either positive or negative. The scope feature determines the possible quantifier scopings of the quantifier arguments, that is, either subject-wide or object-wide. The feature uniq specifies whether a uniqueness constraint must be applied. Some of the features occur in several parts of the control feature. For example, the reading feature occurs in both subject and object features within the control feature and determines the verbal reading to be applied to each argument. As mentioned in the previous section, the argument representations may well influence the particular values of these features. In particular, it would seem reasonable to require of a particular unification grammar devised to construct these feature-matrices that the uniq features in the subject and object control complex features obtain their values from the subject argument control feature uniq and the object argument control feature uniq.

The interpretation of the arguments to a transitive verbal predicate will have derived a denotation graph containing at least two vertices which describe the witness sets of each argument generalized quantifier. That is, in general terms, after the analysis of the arguments to a transitive

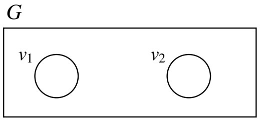


Figure 4.6: The graph derived after the arguments to the transitive verbal predicate given in (200) have been analysed.

verbal relation a graph will have been derived which for (200) would be of the form shown in figure 4.6, where the vertex v_1 is derived from the analysis of *every farmer* and the vertex v_2 has been derived from the analysis of *a donkey*. The interpretation of a verbal predicate centres around the translation of the information within the control feature into a verbal constraint which can be applied to the denotation sets of the arguments to the verbal relation to determine the sets of individuals which satisfy the particular verbal reading described by the control features.

As an example, the control feature for the verbal predicate shown in (200) is shown again below along with, in (202) the constraint derived from it. The particular verbal reading given in the control feature is a positive polarity subject and object distributive reading with subject-wide scope and no uniqueness restriction.

(201)
$$\begin{bmatrix} \text{subject} & \begin{bmatrix} \text{reading} & \text{distributive} \\ \text{pol} & + \\ \text{uniq} & - \end{bmatrix} \\ \text{object} & \begin{bmatrix} \text{reading} & \text{distributive} \\ \text{pol} & + \\ \text{uniq} & - \end{bmatrix} \\ \text{predicate} & \begin{bmatrix} \text{pred} & \text{own} \\ \text{pol} & + \\ \text{scope} & \text{subjectwide} \end{bmatrix} \end{bmatrix}$$

(202) $\lambda C_1, C_2, V \exists S_1 \in C_1 : \forall S_2 \subseteq S_1 : |S_2| = 1 \rightarrow \exists S_3 \in C_2 : \forall S_4 \subseteq S_3 : |S_4| = 1 \rightarrow \langle S_2, S_4 \rangle \in V$

We can apply the rule in (202) to the denotation sets derived from the arguments to the verbal relation, i.e., the denotation sets described by the vertices v_1 and v_2 in figure 4.6. To be fully instantiated the rule also requires a relation V. We can supply the value given by the model to the verbal predicate PRED for this relation, i.e., $F(\text{own})$ for the above example. The rule in (202) provides a constraint on the satisfaction of the particular distributive reading. It requires that for any set S_1 within the subject denotation set, for all its singleton sets (S_2) there is a set S_3 in the object denotation set such that for all its singleton sets (S_4), the pairs $\langle S_2, S_4 \rangle$ satisfy the relation V. Collecting all the sets from the two denotation sets which contribute to the satisfaction of the above constraint a new graph G' can be derived from G where the vertex v_1 in G' contains those witness sets from v_1 in G which satisfy the verbal reading and the vertex v_2 in G' contains those witness sets from v_2 in G which satisfy the verbal reading. If the two vertices v_1 and v_2 in G' are not empty then a relational edge will be constructed between them specifying which sets in v_1 and v_2 are related. The resulting graph G' would have the form shown in figure 4.7, with respect to a satisfying model.

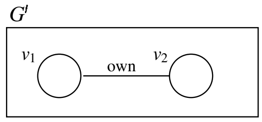


Figure 4.7: The graph derived from the analysis of (200).

4.4.3 Anaphors

Within this section, I will discuss how GTS handles anaphors. In section 4.4.4 anaphor-antecedent relations will be discussed.

The semantic representation provided for a personal pronoun will take the form below.

(203)

$$\left[\text{control} \left[\begin{array}{ll} \text{pred} & \text{PRO} \\ \text{anaphor} & \text{PRO-TYPE} \\ \text{number} & \text{NUM} \end{array} \right] \right]_{pro}$$

The feature variable PRO will take the value of the particular pronoun, e.g., he,she etc. The feature variable PRO-TYPE will take the value either bound or referential while the feature number will provide the syntactic number of the anaphor.

Anaphoric antecedents are described by vertices in particular denotation graphs within a discourse space. In interpreting a pronominal semantic representation we must choose one or more vertices within denotation graphs held within the discourse space derived from the analysis of the previous discourse. From these antecedents a vertex for the anaphor is constructed. The denotation set for the anaphor vertex will be derived from the denotation sets of the antecedent vertices.

The interpretation of referential and bound pronouns differs. I will discuss each in turn. Both, however, follow the basic interpretational process of extending the graph derived from the previous analysis of the sentence in which they occur.

Referential Pronouns

An example discourse with a referential pronoun is shown below.

(204) Every farmer owns a donkey. They are happy.

If we assume that the pronoun *they* refers to the farmers described by the first sentence, then if we are to treat this pronoun referentially, we are only interested in the individuals identified as the farmers in the first sentence. In particular, we are not interested in the relationships these farmers have with certain donkeys described in the first sentence.

The interpretation of a referential pronoun is shown diagrammatically in figure 4.8. The figure shows the incremental extension of a graph containing a vertex v_5. A new vertex v_6 is the vertex provided for the pronoun. The denotation set for this vertex will be derived from some denotation sets in the discourse space.

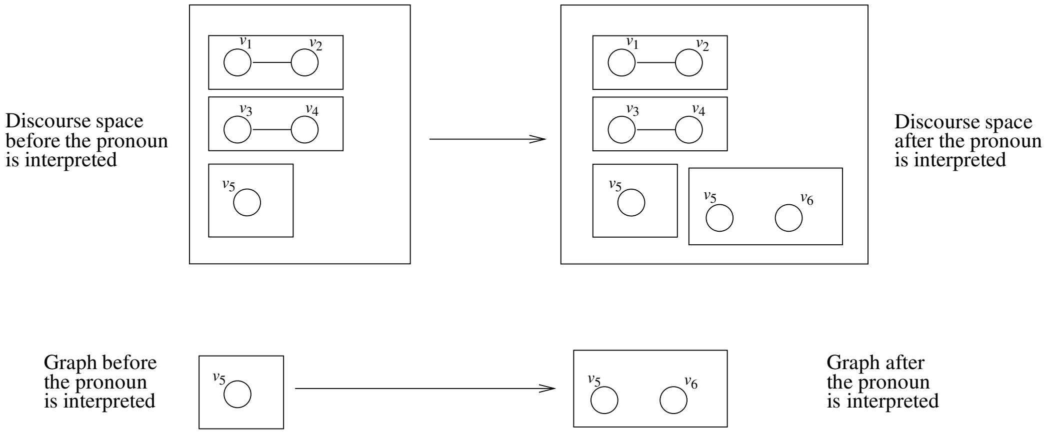


Figure 4.8: Interpretation of a referential pronoun.

Bound Pronouns

An example discourse with a bound pronoun is shown below.

(205) Every farmer owns a donkey. They beat them.

I will assume that the pronoun *they* refers to the farmers identified in the first sentence and the pronoun *them* refers to the donkey's identified in the first sentence. If these pronouns are to be treated in a bound manner then we are not only interested in who the individual farmers and donkeys are from the first sentence but how they relate to each other. This information will be used to ensure that the second sentence when interpreted constrains farmers to only beat donkeys they own.

Figure 4.9 shows the interpretation of a bound pronoun. The figure illustrates an example where a pronoun, whose derived vertex is v_6, is treated as referring to the antecedent vertices v_1 and v_3. Both these antecedent vertices are from different graphs. However, unlike referential pronouns where the individuals from these vertices are all that matters, copies of the entire graphs in which these vertices appear are incorporated into the graph constructed from the interpretation of the pronoun. Anaphoric edges are then constructed between the anaphor and (its copied) antecedents.

4.4.4 Anaphor-Antecedent Relations

In chapter 2 in section 2.2.2, it was shown that the different readings provided for donkey sentences essentially revolved around the different anaphor-antecedent relations that can be provided. Furthermore, in determining the anaphor-antecedent relations, of primary importance was how verbal relations were analysed. This lead me to propose that the treatment of different anaphor-antecedent relations should be centered within the analysis of verbal relations. The table I used to illustrate this is repeated again in (207) which looks at the possibilities open for the analysis of the verbal relations in (206).

(206) Every farmer who owns a donkey beats it.

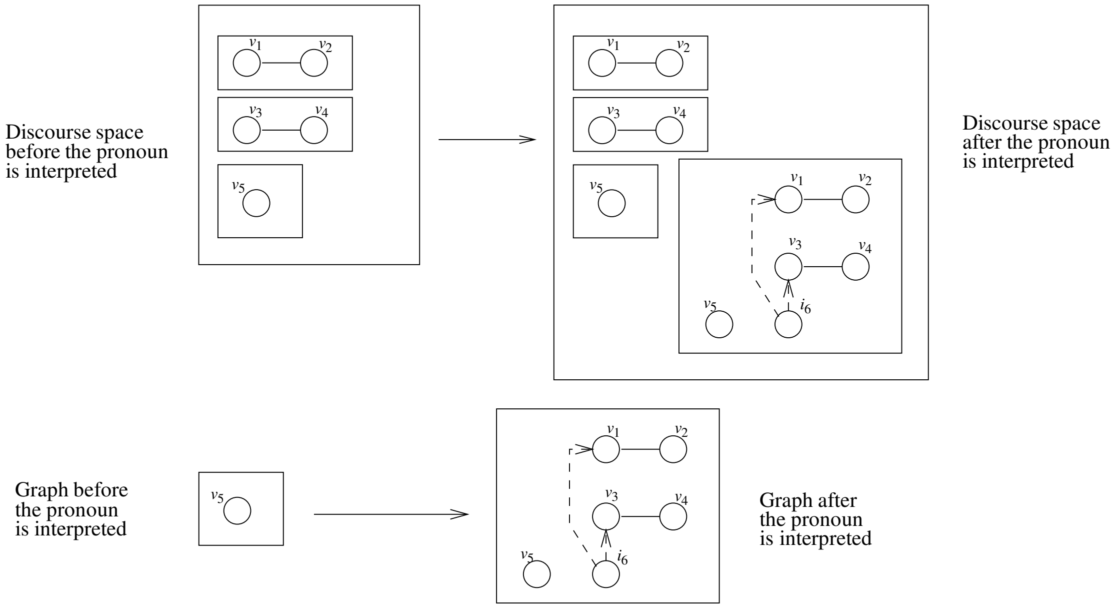


Figure 4.9: Interpretation of a bound pronoun.

(207)

x Donkeys Owned	*y Donkeys Beaten*	*Reading*
$x = 1$	$y = 1$	Unique Antecedent
$x \geq 1$	$y = 1$	Unique Anaphor
$x \geq 1$	$1 \leq y \leq x$	Weak (Indefinite Lazy)
$x \geq 1$	$y = x$	strong (Universal)

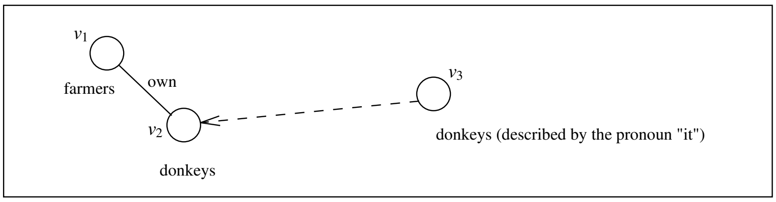
Each particular anaphor-antecedent reading is derived by placing certain constraints on the analysis of the verbal relations in (206). The *unique antecedent* and *unique anaphor* readings require uniqueness constraints to be imposed on either the antecedent's verbal relation or the anaphor's verbal relation, after which either a *weak* or *strong* anaphor-antecedent relation can be imposed. I will leave the discussion concerning the imposition of uniqueness constraints within the analysis of verbal relations to the next chapter. This leaves the imposition of either a *weak* or *strong* anaphor-antecedent relation during the analysis of a verbal relation. The *weak* and *strong* anaphor-antecedent relations are related. When checking whether two sets of individuals satisfy a transitive verbal relation, for both *weak* and *strong* anaphor-antecedent relations we must check not only that the pair of sets satisfies the verbal predicate but also that they are anaphorically acceptable. In analysing the beat verbal predicate within the donkey sentence (206) this would mean that given a farmer and a donkey we must check not only that the farmer beats the donkey but also that she owns it. For a *strong* anaphor-antecedent relation, however, we need to further to check that every farmer beats every donkey she owns.

The main problem then is how to check whether two arguments to a verbal relation are anaphorically acceptable. GTS provides an original method of determining this, as will be explained in the next section.

Denotation Graphs as Constraint Networks

In order to determine whether particular sets chosen from argument vertices to a verbal relation are acceptable, GTS utilizes denotation graphs as **constraint networks**. The denotation graph derived (within a satisfying model) prior to the analysis of the beat verbal relation for the donkey sentence in (208) is shown in (209).

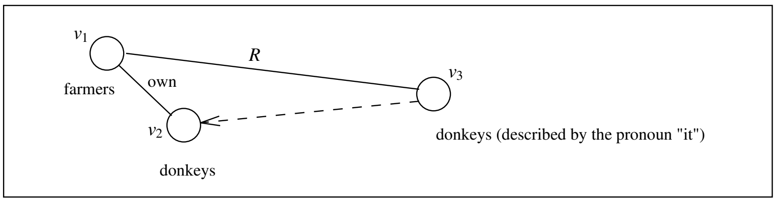
(208) Every farmer who owns a donkey beats it.



(209)

The analysis of the verbal predicate beat will concern the vertices v_1 and v_3. The analysis will derive a relational edge between v_1 and v_3. The addition of this edge to the graph in (209) will produce a *circuit* through the vertices v_1, v_2, v_3. It is the creation of circuits in a denotation graph derived from the analysis of a verbal relation that signifies the existence of bound anaphoric situations.

The analysis of the beat verbal predicate will involve choosing sets from v_1 and v_3 and checking they satisfy the desired verbal reading. However, we wish to ensure that only anaphorically acceptable sets can be satisfied in the analysis of a verbal relation. This can be accomplished by treating the denotation graph as a constraint network. Prior to analysing the appropriate verbal rule, we derive a temporary denotation graph by extending the graph in (209) with a relational edge between v_1 and v_3 with an unrestricted relational constraint given by R, relating every set in v_1 with every set in v_3. The derived graph is shown in (210).



(210)

This graph will be treated as a constraint network which describes a constraint satisfaction problem. A solution to a constraint satisfaction problem of this form is a labelling of each vertex with a set from that vertex, such that the labelling as a whole satisfies all the constraints of the constraint network. From this informal description, two things need to be determined.

1. What do we mean by a label?
2. What are the constraints of the network?

A label for each vertex will be some subset of a set within the denotation set described by that vertex. That is, each vertex describes a set of sets of individuals; a (possible) label will be any subset from any of these sets. Two types of constraint will be defined, one for the relation edges and one for the anaphoric edges in the graph. These constraints are given below, the anaphoric constraint only in its preliminary form.

Definition 9: Relational Edge Constraint.
Given a relational edge $\langle v, v', R\rangle$ where the labels for v and v' are S and S', respectively, then it must be that $\langle S, S'\rangle \in R$.

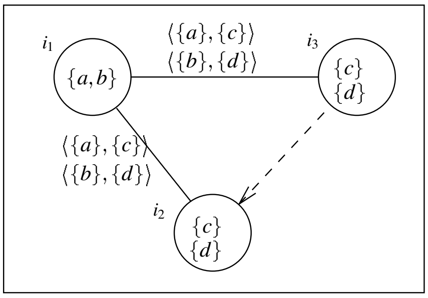
Definition 10: Anaphoric Edge Constraint. (Preliminary Version)
Given an anaphoric edge $\langle v, v'\rangle$, where the labels for v and v' are S and S', respectively, then it must be that $S = S'$

A labelling of the graph which satisfies all the edge constraints is called a globally consistent labelling or a globally satisfiable labelling.

If we determine all possible globally consistent labellings for the graph in (210) we can extract which pairs of labels satisfy v_1 and v_3, i.e., which pairs of labels from v_1 and v_3 contribute to a global satisfaction of the constraint network given in (210). We can then use these sets to limit the analysis of the verbal reading of the original graph, shown in (209). By this means we can ensure that a weak anaphor-antecedent relation is imposed on the verbal analysis of the beat verbal relation.

An example denotation graph is shown in (211) with respect to some model whose details need not be given.

(211)



There are two possible global satisfiable labellings for this graph when it is treated as a constraint network. They are given below as a list of vertex-label pairs.

- $\langle i_1, \{a\}\rangle, \langle i_2, \{c\}\rangle, \langle i_3, \{c\}\rangle$
- $\langle i_1, \{b\}\rangle, \langle i_2, \{d\}\rangle, \langle i_3, \{d\}\rangle$

For a strong anaphor-antecedent relation, we impose a further constraint after the verbal reading has been applied. We check that any subject argument which has satisfied the verbal reading (with the imposed weak anaphor-antecedent relation) has accepted all anaphorically acceptable object arguments.

4.5 Review

Some of the central points made in this chapter are given below.

- A set-theoretic model structure is utilized.

- The separation of lexical nouns, which derive zero-place predicates, from intransitive verbs, which describe one-place predicates taken along with the lack of traditional variables in the semantic representation allows a uniform analysis to be given to collective and distributive verbal readings.

- A graph-theoretic denotational structure (a *denotation graph*) is used to hold discourse information. These structures are collected together into a *discourse space*.

- Antecedents are identified with particular vertices within a denotation graph.

- The semantic representational structure is described by unification feature matrices. A form of representation common in present-day grammar formalisms.

- Noun phrases are interpreted as generalized quantifiers.

- The feature-based representational language allows a variety of interpretational constraints to be straightforwardly expressed.

- The analysis of anaphors is separated from the analysis of the resulting anaphor-antecedent relations, the latter being interpreted within the analysis of verbal relations.

- The determination of when two arguments to a verbal relation are anaphorically acceptable is determined by viewing the denotation graph from which the arguments are taken as a constraint network and solving the constraint satisfaction problem posed by this network.

- Both strong (universal) and weak (indefinite lazy) anaphor-antecedent readings are available, as well as readings requiring uniqueness constraints, i.e., unique anaphor and unique antecedent readings.

- Anaphoric information derived from sub-sentential linguistics constituents is available.

Chapter 5

A Semantic Framework for Non-anaphoric Discourse

Having provided an overview of the GTS framework in the previous chapter, the next two chapters will lay out the framework in detail. Within this chapter, I will discuss and provide a semantics for a small subset of non-anaphoric English discourse. The next chapter will extend the semantics into anaphoric discourse to cover pronominal noun phrase anaphora.

I will begin by discussing in a more formal manner the consistent labelling of a denotation graph when it is treated as a constraint network. Following this, I shall look at the semantic interpretation of different semantic predicates associated with particular linguistic constituents. For each particular predicate type I shall discuss the semantic representation provided, explaining the particular significance of each feature. I shall then provide a detailed exposition of the semantic interpretation for this type of predicate structure. As has been mentioned previously, the framework will be limited to extensional declarative discourses containing unary generalised quantifiers. Syntactically, I have concentrated on covering enough syntactic constructions to handle quantified donkey sentences and simple negatives. In section 5.3, I will discuss lexical nouns and proper names, the constituents which introduce individuals into a discourse. Following this, I will look at generalised quantifiers, fleshing out the details of the treatment proposed in the previous chapter. Verbal relations will be discussed in section 5.3.3. In section 5.4 I will look at the determination of truth with respect to an interpretation. In section 5.5, I will look at the analysis of non fully-instantiated semantic predicates, i.e., semantic predicates with uninstantiated argument features.

5.1 Notational Conventions

Certain notational conventions will be used by the semantic interpretation rules. These conventions are outlined below.

- The subsumption operation on feature structures will be denoted by the symbol $\sqsubseteq$.
- If v is a vertex and G is a graph then $v \in G$ will mean that the vertex v is contained within the vertex set in G.

- $G[v'_1/v_1, v'_2/v_2, \ldots, v'_n/v_n]$ will mean the graph G' which is exactly the same as G *except* that each vertex $v'_i(1 \leq i \leq n)$ replaces the vertex v_i where ever v_i occurs in G. Sometimes it will be useful to describe the changes in vertices as a function. Given a function f which maps each vertex in G to a new vertex, $G[f]$ will mean the same as above except each vertex v is transformed to $f(v)$.
- $G \cup G'$ will mean the component-wise union of the graphs G and G'.
- $G[v]$ where v is a vertex will mean the graph derived by adding the vertex v to the vertex set in G.
- $G[e]$ where e is a relational edge will mean the graph derived by adding the relational edge e to the relational edge set in G.
- If α is a feature structure then $\alpha/Path$ will be the value of the feature identified by the feature path $path$ in α. A feature path, $path$ from α will be given as $\langle f_1, f_2, \ldots, f_n \rangle$ where $f_i(1 \leq i \leq n)$ is some feature in α.

5.2 Consistent Labelling of a Graph

When denotation graphs are treated as constraint networks the notion of a consistent labelling of the graph is used to help determine the correct analysis of anaphor-antecedents relations. The concept of a consistent labelling is also utilized in order to describe the notion of a maximally consistent graph, which will be formalised at the end of this section.

Cooper, Cohen and Jeavons (1994) describe a finite constraint satisfaction problem (with binary constraints) as follows:

> A *finite constraint satisfaction problem* (CSP) consists of a finite set of nodes, N (identified by the natural numbers $1, 2, \ldots, n$), each of which has an associated finite set of possible labels A_i. The labellings allowed for specified pairs of nodes are restricted by a set of constraints, C. Each constraint $C_{ij} \in C$ is a list of pairs of labels from A_i and A_j which may be simultaneously assigned to the nodes i and j, i.e., $C_{ij} \subseteq A_i \times A_j$. A solution to a CSP is a labelling of the nodes which is consistent with all the constraints.

In the case of denotation graphs the nodes of the CSP will be the vertices of the graph. The definition of a *label* is defined below.

Definition 11: A label is a set of individuals (from a model).

We will need to label vertices in a graph and, therefore, next the definition of a label for a vertex is provided.

Definition 12: A label for a vertex $v = \langle i, \mathcal{C} \rangle$ is $\{\}$ if $\mathcal{C} = \{\}$ otherwise it is some label S such that $\exists X \in \mathcal{C} \wedge S \subseteq X$.

A label for a vertex is therefore some subset of some set within the denotation set of that vertex, or the empty set if the vertex is empty. Next, we can define a *labelling* for a graph.

Definition 13: A labelling $\mathcal{L}$ for a graph G is a function mapping a label to each vertex in G.

A consistent labelling of a graph (i.e., a solution to the CSP described by the denotation graph) is a labelling which satisfies the relational and anaphoric edge constraints. These constraints are repeated below.

Definition 14: Relational Edge Constraint.
Given a relational edge $\langle v, v', R\rangle$ where the labels for v and v' are S and S', respectively, then it must be that $\langle S, S'\rangle \in R$.

Definition 15: Anaphoric Edge Constraint. (Preliminary Version)
Given an anaphoric edge $\langle v, v'\rangle$, where the labels for v and v' are S and S', respectively, then it must be that $S = S'$

I will define a relation *satis* which takes a graph, G, and a labelling $\mathcal{L}$. The relation is satisfied if the labelling $\mathcal{L}$ of G is a consistent labelling.

Definition 16: $satis(G, \mathcal{L})$ iff G is a denotation graph and $\mathcal{L}$ is a consistent labelling of G.

We can utilize the *satis* relation in globally propagating local changes to a denotation graph. If we change the contents of a particular vertex, for example by rejecting certain sets, we want to modify other vertices in the graph to respect this change. In order to do this we must decide what justification is required in retaining a set within an arbitrary vertex. The decision taken is to retain any set within a vertex for which some subset of that set contributes to a consistent labelling. I will begin by defining with respect to a graph G a function J which given a vertex v returns a vertex v', where v' contains only those sets from v which are justified by a consistent labelling.

Definition 17: Given a vertex $v = \langle i, \mathcal{C}\rangle$ in some graph G, $J(v) = \langle i, \mathcal{C}'\rangle$ where $\mathcal{C}' = \{X \in \mathcal{C} | \exists \mathcal{L} : satis(G, \mathcal{L}) \wedge \mathcal{L}(v) \subseteq X\}$

We can use this definition to define a function *cons* which given a graph G returns a graph G' in which all the vertices are maximally consistent.

Definition 18: Given a graph G, then $cons(G) = G[J]$

Definition 17 assumes a set in a denotation set is justified by a consistent labelling if some *subset* of that set is a label for some consistent labelling of the entire graph. An obvious more restrictive alternative is to allow a set to be justified if for some collection of subsets whose union is the entire set each of the sets in the collection is a label for some consistent labelling of the graph. However, in GTS both positive and negative verbal readings derive relational edges of the same type and *some* negative readings are satisfied by relating only some subset of a set within a denotation set with some other set in another vertex. For example, two contrasting sentences are shown below which illustrate the point.

(212) Every farmer owns a donkey.

(213) Every farmer does not own a donkey.

In both examples, the derived vertex for *every farmer* will contain the single set of all farmers. For a standard subject distributive reading of (212) we will require that every singleton set containing an individual farmer from the set of all farmers it is the case that the farmer owns a donkey. However, for a reading of (213) in which we apply sentential negation along with a subject distributive reading we will only require that there is at least one singleton set (from the set of all farmers) containing a farmer such that that farmer does not own a donkey. That is, the derived relationship between the vertices containing the farmers and the donkeys may not involve the entire set of farmers. The empirical predictions of the semantics are not adversely effected by what would seem a watered down form of justification for positive verbal relations to cope with certain negative verbal relations. The reason for this is that the analysis of verbal relations themselves will always require justification via a consistent labelling for anaphorically related arguments ensuring that only valid sets of individuals satisfy any verbal relation. For instance, if we followed (213) with the following sentence:

(214) They hate them.

Assuming in (214) we are trying to say that the farmers from (213) hate the donkeys they don't own, then although the set of all farmers satisfied the interpretation of (214) a positive distributive reading of (214) will not be satisfied by the set of all farmers if only a subset of farmers from the interpretation of (213) are related to donkeys (they don't own). That is, although the semantics allows the set of all farmers to be justified within the graph derived from the interpretation of (213) even though only some subset of that set's farmers are related to donkeys this does not allow incorrect readings to be derived on the set of farmers in later anaphoric discourse.

5.3 Semantic Interpretation

The semantic interpretation will cover lexical noun predicates, determiner predicates, and verbal predicates. The particular semantic interpretation rule to apply to a particular semantic feature representation is determined by testing whether the feature structure in question is subsumed by a template feature structure representing the particular predicate type.

The semantic interpretation function $[\![]\!]$ has the following general specification where α is a formula of the semantic representation language, $\mathcal{D}$ and $\mathcal{D}'$ are input and output discourse spaces, G and G' are input and output denotation graphs, i is a vertex identifier and M is a model, and I is a set of identifiers.

(215) Semantic Interpretation Function

$$[\![\alpha]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G', \mathcal{D}') \rangle$$

The specification requires that the semantic interpretation function takes a pair containing a graph and a discourse space and returns a triple consisting of an identifier, a graph and a discourse space. How the semantic interpretation function is utilized to interpret a discourse is discussed below.

A discourse as has been discussed previously is to be understood as a temporally ordered set of declarative extensional sentences $S_1, S_2, \ldots, S_n$. An appropriate (compositional) analysis will, given a sentence in the discourse, S_j, derive an appropriate feature matrix A_j representing the syntactic and semantic information of S_j. The value of the sem feature of A_j will be α_j, which will contain the semantic representation derived from S_j. That is the analysis of a discourse

$S_1, S_2, ..., S_n$ will produce a series of semantic representations $\alpha_1, \alpha_2, ..., \alpha_n$. The semantic interpretation function will be applied under the temporal ordering to each semantic representation α_j, in the following manner, where M is a model and I is a set of identifiers.

(216) $[\![\alpha_j]\!]^{M,I} = \langle((\langle\{\},\{\},\{\}\rangle, \mathcal{D}_{j-1}), (i, G_j, \mathcal{D}_j)\rangle$

The analysis of each sentence is begun with an empty input graph. The output discourse space derived from the analysis of α_{j-1} is forced to be the input discourse space for the analysis of α_j.

The semantic interpretation provided for the first sentence in a discourse is shown below.

(217) $[\![\alpha_1]\!]^{M,I} = \langle((\langle\{\},\{\},\{\}\rangle, \{\}), (i, G_1, \mathcal{D}_1)\rangle$

In (217), an empty discourse space is provided because within the strict extensional framework that is being devised no anaphoric information is deemed to exist before the beginning of a new discourse. The method in (216) and (217) provides an inductive procedure for the analysis of discourses of any length. This procedure formalizes the intuitive description given by figure 4.5 in the last chapter.

It should be emphasised that the inductive specification of the application of the semantic interpretation function is concerned with sentences in a discourse and thus the input graph is always empty. However, during the application of the interpretation function to semantic predicates contained within the top-level predicate derived for a sentence the input graph may well not be empty. That is, the processing of each sentence begins with an empty graph which is incrementally extended through the analysis of the semantic representation for that sentence.

5.3.1 Lexical nouns

The semantic representation given to lexical nouns will be of the form:

$$\begin{bmatrix} \mathsf{control} & \begin{bmatrix} \mathsf{pred} & \mathsf{PRED} \\ \mathsf{number} & \mathsf{NUM} \end{bmatrix} \end{bmatrix}_n$$

The control feature structure contains at least the two features, pred and number. Other features to possibly help determine the interpretation of verbal predicates will not be discussed as they don't impinge on the denotation given to lexical nouns. The pred feature specifies the particular predicate for the lexical noun in question, e.g., farmer. The feature number gives the syntactic number of the lexical noun and may take the value singular or plural.

A model, $M = \langle D, F\rangle$, will specify via $F(\mathsf{PRED})$ the set of individuals attributable to the lexical noun. From this information two different denotations will be derived for the interpretation of lexical nouns dependent on the value of the number feature. The two possible denotation sets derived for a nominal predicate, PRED, with respect to a model, $M = \langle D, F\rangle$, are as follows.

1. If number = singular then derive $\{X \subseteq F(\mathsf{PRED}) | \ |X| = 1\}$.

2. If number = plural then derive $\{X \subseteq F(\mathsf{PRED}) | \ |X| \geq 1\}$

That is, we collect together all singleton sets of individuals from the set specified by $F(\mathsf{PRED})$ for singular lexical nouns and we collect together all possible sets of individuals from the set specified by $F(\mathsf{PRED})$ for plural lexical nouns.

The exact semantic interpretation for a lexical noun predicate is given below, where M is a model, I is a set of identifiers and α is a feature-based semantic representation.

If $\left[\text{control} \left[\begin{array}{ll} \text{pred} & \text{A} \\ \text{number} & \text{B} \end{array} \right] \right]_n \sqsubseteq \alpha$ and $\left(\begin{array}{l} \text{PRED} = \alpha/\langle \text{control pred} \rangle \\ \text{NUM} = \alpha/\langle \text{control number} \rangle \end{array} \right)$ then
$[\![\alpha]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G[v], \mathcal{D} \cup \{G[v]\}) \rangle$, where,

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$.
- $v = \langle i, C \rangle$ where $C = \begin{cases} \{X \subseteq F(\text{PRED}) \mid |X| = 1\} & \text{If NUM} = \text{singular} \\ \{X \subseteq F(\text{PRED}) \mid |X| \geq 1\} & \text{If NUM} = \text{plural} \end{cases}$

5.3.2 Generalized Quantifiers

Within this section, I shall discuss the formal interpretation given to generalized quantifiers which semantically encompass the syntactic class of noun phrases. I have already given quite a detailed overview in the last chapter in section 4.4.1 and I will therefore begin by briefly reviewing the pertinent points.

Denotations for (unary) generalized quantifiers are derived during the process of analysing determiner predicates. The semantic representation for a determiner predicate is given below.

$$\left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll} \text{pred} & \text{PRED} \\ \text{number} & \text{NUM} \\ \text{uniq} & \text{UNIQ} \\ \text{pol} & \text{POL} \end{array} \right] \\ \text{arg1} & \text{ARG1} \end{array} \right]_{det}$$

The control feature structure contains several features. The pred feature contains the particular determiner predicate in question, for example every or two. The number feature determines the (syntactic) number. Finally, there are two features (uniq and pol) which play an important role in determining the correct reading quantifiers have when they are involved with verbal relations. These features don't determine in any way what denotation is provided for the generalized quantifier but as they are important for understanding the analysis of generalized quantifiers in relation to verbal relations they are included within the discussion. Furthermore, they are included as any appropriate construction of the semantic representations in GTS within a particular unification grammar would wish to identify appropriate values for the uniq and pol features in determiner predicates in order to transfer these values to the appropriate control features of verbal predicates whose arguments are determiner predicates. The uniq feature can take the value + or − and represents whether there should be a uniqueness restriction applied to the analysis of any verbal relation in which the denotation of the quantifier is applied. The pol feature can take the value + or − and represents whether the quantifier contributes positively or negatively to any verbal relation to which it is involved.

The denotation set derived for a determiner predicate is dependent on two important results of generalized quantifier research originally discussed in Barwise and Cooper (1981). That is, the "lives on" or conservativity universal and the idea of a witness set for a quantifier. The definitions of these two concepts, already discussed in the last chapter, are repeated below, where D is the domain of individuals specified by some model, A and B are sets of individuals and Det is a determiner, thus making $Det_D A$ a generalized quantifier over domain D.

(218) *Conservativity Universal*: $Det_D AB \Leftrightarrow Det_D A(A \cap B)$ (Barwise & Cooper, 1981, p. 178)

(219) A *witness set* for a quantifier $Det_D A$ living on A is any subset w of A such that $w \in Det_D A$. (Barwise & Cooper, 1981, p. 191).

The conservativity universal (assuming it is correct) allows us to concentrate only on those individuals that are introduced by lexical nouns (and proper names) when determining the interpretation for generalized quantifiers. The idea of a witness set will be utilized when deriving a denotation set describing a generalized quantifier. All those witness sets that satisfy the generalized quantifier will be utilized to construct the new denotation set. The particular use for the witness sets of a quantifier will depend on a further characterisation given by Barwise and Cooper termed monotonicity. The definitions, of monotone increasing, monotone decreasing and non-monotone quantifiers are repeated below.

- A quantifier Q is *monotone increasing* if $X \in Q$ and $X \subseteq Y \subseteq D$ implies $Y \in Q$.
- A quantifier Q is *monotone decreasing* if $X \in Q$ and $Y \subseteq X \subseteq D$ implies $Y \in Q$.
- A quantifier A is non-monotone if it is neither monotone increasing nor monotone decreasing.

For monotone increasing and non-monotone quantifiers the denotation set derived by the semantics will contain all those witness sets (as defined by Barwise and Copper) which are applicable to the quantifier. For monotone decreasing quantifiers the denotation set derived will contain all those sets which are complements (with respect to A, the set of individuals that satisfies the argument to the quantifier) of the witness sets applicable to the quantifier. This interpretation of monotone decreasing quantifiers is related to the interpretation of monotone decreasing quantifiers given by van den Berg (1993) who interprets them as the negation of monotone increasing quantifiers. When interpreting a noun phrase in GTS, a denotation set $\mathcal{C}$ is derived, where $\mathcal{C}$ is determined as below for particular types of quantifiers, $Det_D A$, where D is the model domain and A is the maximal set of individuals that satisfy the nominal argument to the determiner Det.

(220) If $Det_D A$ is monotone increasing or non-monotone then $\mathcal{C} = \{w | w \in Det_D A\}$

(221) If $Det_D A$ is monotone decreasing then $\mathcal{C} = \{A - w | w \in Det_D A\}$

The set A of individuals satisfying the nominal phrase is determined from the analysis of the argument to the determiner predicate. This analysis will have derived a graph, one of whose identified vertices describes the individuals satisfying the analysis of the nominal argument. We will need to take the union of these sets of individuals identified by this vertex to obtain the set A. From this set we can determine the witness sets that satisfy the quantifier.

I shall now provide the formal semantic interpretation of a determiner predicate, where M is a model, I is a set of identifiers and α is a feature-based semantic representation.

$$\text{If} \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{A} \\ \textsf{number} & \textsf{B} \\ \textsf{uniq} & \textsf{C} \\ \textsf{pol} & \textsf{D} \end{bmatrix} \\ \textsf{arg1} & \textsf{G} \end{bmatrix}_{det} \sqsubseteq \alpha \text{ and } \begin{pmatrix} \textsf{PRED} = \alpha/\langle \textsf{control pred} \rangle \\ \textsf{NUM} = \alpha/\langle \textsf{control number} \rangle \\ \textsf{ARG} = \alpha/\langle \textsf{arg1} \rangle \end{pmatrix} \text{ then}$$

$[\![\alpha]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G'', \mathcal{D}' \cup \{G''\}) \rangle$ where

- $[\![\textsf{ARG}]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G', \mathcal{D}') \rangle$
- $\langle i, \mathcal{C} \rangle \in G'$, $S = \bigcup_{X \in \mathcal{C}} X$
  The vertex holding the information of the argument to the determiner is $\langle i, \mathcal{C} \rangle$. We take the union of the sets in $\mathcal{C}$.
- $$\mathcal{C}' = \begin{cases} \{X \subseteq S | X = S\} & \text{If } \textsf{PRED} = \textsf{every} \\ \{X \subseteq S | |X| \geq \frac{1}{2}|S|\} & \text{If } \textsf{PRED} = \textsf{most} \\ \{X \subseteq S | |X| = 1\} & \text{If } \textsf{PRED} = \textsf{a} \\ \{X \subseteq S | |X| = 1\} & \text{If } \textsf{PRED} = \textsf{some}, \textsf{NUM} = \textsf{singular} \\ \{X \subseteq S | |X| > 1\} & \text{If } \textsf{PRED} = \textsf{some}, \textsf{NUM} = \textsf{plural} \\ \{S - X | X = \{\}\} & \text{If } \textsf{PRED} = \textsf{no} \\ \{S - X | X \subseteq S \wedge |X| < \frac{1}{2}|S|\} & \text{If } \textsf{PRED} = \textsf{few} \\ \{X \subseteq S | |X| = 2\} & \text{If } \textsf{PRED} = \textsf{two} \end{cases}$$
  From the set S we can determine the denotation set $\mathcal{C}'$ containing the witness sets for the particular quantifier in question.
- $G'' = G'[\langle i, \mathcal{C}' \rangle / \langle i, \mathcal{C} \rangle]$
  The graph G'' is the graph G' with the vertex $\langle i, \mathcal{C} \rangle$ replaced by the vertex $\langle i, \mathcal{C}' \rangle$.

The interpretation of the determiner *few* might not conform to some reader's intuitions about this determiner. The present interpretation treats *few* as *not most*. An alternative interpretation would be to state that *few Xs* identifies some sets of small numbers of *Xs* or sets of less than half *Xs*, which ever is greater. An alternative interpretation rule which could be substituted for the one given is shown below. I have arbitrarily chosen 5 as a typical small number.

- $\mathcal{C}' = \{S - X | X \subseteq S \wedge |X| < \frac{1}{2}|S| \vee |X| < 5\}$ If $\textsf{PRED} = \textsf{few}$

I will end this section by providing the interpretation of proper names which are also treated as generalized quantifiers. Proper names introduce new vertices unlike the other generalized quantifiers discussed above. The denotation set provided within their vertex contains a single set holding the individuals identified by the model for the particular proper name. The semantic interpretation rule is given below where M is a model, I is a set of identifiers and α is a feature-based semantic representation.

If $\begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{A} \end{bmatrix} \end{bmatrix}_{pn} \sqsubseteq \alpha$ and $\textsf{PRED} = \alpha/\langle \textsf{control pred} \rangle$ then
$[\![\alpha]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G[v], \mathcal{D} \cup \{G[v]\}) \rangle$, where

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$.
- $v = \langle i, \mathcal{C} \rangle$ where $\mathcal{C} = \{F(\textsf{PRED})\}$

$$
\begin{bmatrix}
\text{control} & \begin{bmatrix}
\text{subject} & \begin{bmatrix} \text{number} & \text{NUM1} \\ \text{reading} & \text{SREADS} \\ \text{pol} & \text{POLS} \\ \text{negreading} & \text{NRS} \\ \text{uniq} & \text{USS} \end{bmatrix} \\
\text{object} & \begin{bmatrix} \text{number} & \text{NUM2} \\ \text{reading} & \text{SREADO} \\ \text{pol} & \text{POLO} \\ \text{negreading} & \text{NRO} \\ \text{uniq} & \text{USO} \end{bmatrix} \\
\text{predicate} & \begin{bmatrix} \text{pred} & \text{PRED} \\ \text{scope} & \text{SCOPE} \\ \text{pol} & \text{POLV} \\ \text{negreading} & \text{NRV} \\ \text{aarel} & \text{AAREL} \end{bmatrix}
\end{bmatrix} \\
\text{arg1} & \text{ARG1} \\
\text{arg2} & \text{ARG2}
\end{bmatrix}_{tv}
$$

Figure 5.1: The semantic feature structure provided for a transitive verbal predicate.

5.3.3 Transitive Verbal Relations

I shall deal with transitive verbal relations in this section. I have already given an overview of verbal relations in section 4.4.2 of the last chapter. I shall begin by discussing in more detail the particular feature structure given to a transitive verbal predicate. In particular, I shall concentrate on the makeup of the control feature for verbal predicates which is used by the semantic interpretation to determine the particular verbal reading to be applied.

The feature structure provided for a transitive verbal predicate is shown in figure 5.1. I will now describe the particular significance of each of the features in the control complex feature. I will use the notation ⟨subject reading⟩ when discussing the feature reading within the subject complex feature, along with related conventions for the object and predicate complex features.

The features ⟨subject reading⟩ and ⟨object reading⟩ determine the particular type of verbal reading to be given. The possible values for these features are given below.

Feature	*Possible Values*
⟨subject reading⟩	distributive,collective1,collective2
⟨object reading⟩	distributive,collective1,collective2

The value distributive determines a distributive reading. The value collective1 determines the standard collective reading: the C_1 reading as defined by van der Does (1991). The value collective2 determines the alternative collective reading proposed by Scha (1981). These readings were discussed in section 2.3.2 of chapter 2.

The features ⟨subject pol⟩, ⟨object pol⟩ and ⟨predicate pol⟩ determine the polarity of the subject argument, object argument and verbal predicate, respectively. The features ⟨subject negreading⟩, ⟨object negreading⟩ and ⟨predicate negreading⟩ describes the type of negative reading, given negative polarity. The values for these features are given below.

Feature	*Possible Values*
⟨subject pol⟩,⟨object pol⟩, ⟨predicate pol⟩	$+,-$
⟨subject negreading⟩,⟨object negreading⟩, ⟨predicate negreading⟩	s,vp,v

The negreading features have three values, s for sentence negation, vp for verb phrase negation and v for verb negation. The three types of negative reading can be illustrated by an example.

(222) Every farmer does not own a donkey.

Under sentence negation (222) is read as saying that it is not the case that every farmer owns a donkey, i.e., some farmer exists who does not own a donkey. Under verb phrase negation (222) is read as saying that every farmer owns no donkeys, while under verb negation, (222) is read as saying that for every farmer there is some donkey that he does not own[1]. Sentences with multiple negative elements will be discussed later.

The feature ⟨predicate scope⟩ determines the scope of each noun phrase within the verbal reading. The possible values for this feature are given below.

Feature	*Possible Values*
⟨predicate scope⟩	subjectwide,objectwide

The different possibilities can be seen in the simple example below.

(223) Every farmer owns a donkey.

For the verbal predicate own with a ⟨predicate scope⟩ feature set to objectwide along with other features set to provide a positive polarity no uniqueness restriction distributive reading (of both noun phrases) the reading obtained is that every farmer owns a particular donkey. To be precise, every donkey owned by a farmer is owned by every farmer. To obtain the reading where every farmer owns only a single particular donkey, we need a uniqueness restriction, as discussed below. If the ⟨predicate scope⟩ feature were set to subjectwide the reading obtained (with no uniqueness restriction) would be where every farmer owns some donkey, not necessarily the same donkey for all farmers.

The features ⟨subject uniq⟩ and ⟨object uniq⟩ determine whether any uniqueness restriction needs to be placed on the subject or object noun phrases. The possible values for these features are given below.

Feature	*Possible Values*
⟨subject uniq⟩	$+,-$
⟨object uniq⟩	$+,-$

These features interact with the ⟨predicate scope⟩ feature to provide a wide range of readings. I will look again at the sentence in (223). A uniqueness restriction requires that a single witness set only from either the subject or object argument be satisfied as part of the verbal reading. Assuming a positive polarity distributive reading, the number of readings available by utilizing the features ⟨predicate scope⟩, ⟨subject uniq⟩, ⟨object uniq⟩ is six. However, whether a

[1]There are other possible readings of (222) obtained by placing stress on particular linguistic constituents. For example, placing stress on the verb *own* we get a reading which says that every farmer is in some relation to a donkey but the relation is not of ownership. These and other stress related readings I will not be dealing with.

Features	No. donkeys owned by each farmer	No. donkeys owned by all farmers	Each donkey owned by all farmers
$\langle$predicate scope$\rangle$: subjectwide $\langle$object uniq$\rangle$: $-$	≥ 1	≥ 1	not required
$\langle$predicate scope$\rangle$: subjectwide $\langle$object uniq$\rangle$: $+$	1	≥ 1	not required
$\langle$predicate scope$\rangle$: objectwide $\langle$object uniq$\rangle$: $-$	≥ 1	≥ 1	required
$\langle$predicate scope$\rangle$: objectwide $\langle$object uniq$\rangle$: $+$	1	1	required

Table 5.1: Different readings of (223) due to values of $\langle$predicate scope$\rangle$ and $\langle$object uniq$\rangle$.

uniqueness restriction is applied to the subject argument or not makes no difference as the denotation of *every farmer* will always contain only a single witness set[2]. Therefore, in reality only the $\langle$predicate scope$\rangle$ and $\langle$object uniq$\rangle$ features produce different readings for this sentence. The different possible readings are summarised in table 5.1. Under a $\langle$predicate scope$\rangle$: subjectwide, $\langle$object uniq$\rangle$: $-$ reading we get the reading where every farmer owns one or more donkeys not necessarily the same for each farmer. Under a $\langle$predicate scope$\rangle$: subjectwide, $\langle$object uniq$\rangle$: $+$ reading we get the reading where every farmer must only own one donkey each. Under a $\langle$predicate scope$\rangle$: objectwide, $\langle$object uniq$\rangle$: $-$ reading we get the reading where every donkey that is owned by a farmer is owned by all of them. Under a $\langle$predicate scope$\rangle$: objectwide, $\langle$object uniq$\rangle$: $+$ reading we get the reading where every farmer owns one particular donkey, that being the single donkey owned by all farmers.

The feature $\langle$predicate aarel$\rangle$ determines the type of anaphor-antecedent relation, i.e., weak (indefinite lazy) or strong (universal). The set of possible values for this feature is given below.

Feature	*Possible Values*
aarel	weak,strong

I will now discuss the interpretation of transitive verbal predicates. The analysis of the arguments to a verbal relation will determine a graph two of whose vertices will describe the subject and object arguments to the relation. Let us assume these two vertices are $v = \langle i, \mathcal{C}\rangle$ and $v' = \langle j, \mathcal{C}'\rangle$. We wish to determine which witness sets from $\mathcal{C}$ and $\mathcal{C}'$ satisfy the verbal reading described by the control features. For each possible verbal reading described by the control features the interpretation will derive a rule which will specify the requirements for sets from $\mathcal{C}$ and

[2]The empty set if there are no farmers.

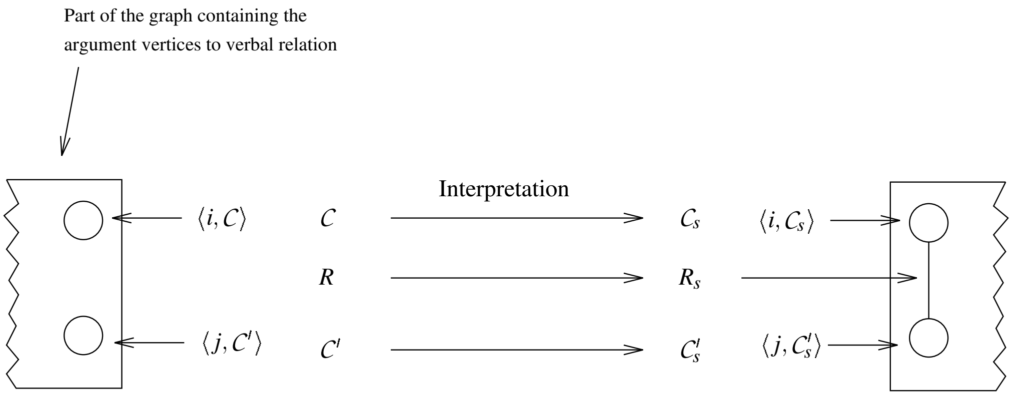


Figure 5.2: A viewpoint on the analysis of a transitive verbal predicate.

C' to satisfy the verbal reading. If any sets in C and C' satisfy the verbal reading we will also wish to construct a relational edge within the resulting denotation graph. This edge will specify which elements (subsets of sets) of C and C' are related. That is, the situation is as shown in figure 5.2. From the denotation sets of the two arguments, C and C' we wish to construct new denotation sets C_s and C'_s which contain those sets from C and C', respectively, which satisfy the particular verbal reading. We will also derive a relation R_s specifying which elements (subsets of sets) in C_s and C'_s are related, as determined by the verbal reading. However, in viewing C_s as derived from C and C'_s as derived from C' we can view R_s as being derived from a relation R, the relation R relating every subset of a set in C with every subset of a set in C'.

Given that the argument denotation sets C and C' to the verbal relation were taken from vertices $v = \langle i, C \rangle$ and $v' = \langle j, C' \rangle$, the new graph constructed will contain the vertices $\langle i, C_s \rangle$ in place of v and $\langle j, C'_s \rangle$ in place of v'. A relational edge will be derived (if $R_s \neq \{\}$), the edge being described by $\langle \langle i, C_s \rangle, \langle j, C'_s \rangle, R_s \rangle$.

The particular verbal reading described by the control information determines the rule which helps determine the mapping between $\langle C, C', R \rangle$ and $\langle C_s, C'_s, R_s \rangle$. I will show how the interpretation rule providing the appropriate constraint on this mapping can be derived compositionally from the information within the control feature. I will derive three interpretation rules: a *subject rule* derived from the information in the $\langle$control subject$\rangle$ complex feature; an *object rule* derived from the information in the $\langle$control object$\rangle$ complex feature; and a *verbal rule* derived from information in the $\langle$control predicate$\rangle$ complex feature. These rules can be specified utilizing the lambda calculus in order that they can be combined compositionally to form a complete rule describing the constraint on the verbal reading.

For uniqueness restrictions, we need to check that only a single witness set satisfies the verbal reading. For this purpose we need to use definite existential quantifiers, as defined below.

Definition 19: Definite Existential Quantifier.

$$\exists! X\phi \equiv \exists X\phi \wedge [\forall Y\phi \rightarrow X = Y]$$

Interpretation rules can be derived describing the readings provided to the subject and object noun phrases. The features $\langle$subject reading$\rangle$ and $\langle$object reading$\rangle$ along with the uniqueness features $\langle$subject uniq$\rangle$ and $\langle$object uniq$\rangle$ can be treated together. Given either the subject or object control features, the *subject rule* or *object rule* derived is shown below, where the features reading and uniq are meant to apply either to ⟨subject reading⟩ and ⟨subject uniq⟩ or to ⟨object reading⟩ and ⟨object uniq⟩.

Feature : Value	Subject or Object Rule
reading : distributive uniq : $-$	$\lambda P, C_1.\exists S_1 \in C_1 : \forall x \in S_1 \rightarrow P(\{x\})$
reading : distributive uniq : $+$	$\lambda P, C_1.\exists! S_1 \in C_1 : \forall x \in S_1 :\rightarrow P(\{x\})$
reading : collective1 uniq : $-$	$\lambda P, C_1.\exists S_1 \in C_1 : P(S_1)$
reading : collective1 uniq : $+$	$\lambda P, C_1.\exists! S_1 \in C_1 : P(S_1)$
reading : collective2 uniq : $-$	$\lambda P, C_1.\exists S_1 \in C_1 : \exists C_2 \subseteq \wp(S_1) : \bigcup C_2 = S_1 \wedge \forall S_2 \in C_2 \rightarrow P(S_2)$
reading : collective2 uniq : $+$	$\lambda P, C_1.\exists! S_1 \in C_1 : \exists C_2 \subseteq \wp(S_1) : \bigcup C_2 = S_1 \wedge \forall S_2 \in C_2 \rightarrow P(S_2)$

The distributive reading rule is intended to identify sets, S_1, in a denotation set, C_1, such that all the singleton sets, $\{x\}$, from S_1 satisfy some formula P. The collective1 reading is intended to identify sets, S_1, within a denotation set, C_1, which satisfy some formula P. The collective2 reading, is intended to identify sets, S_1, within a denotation set, C_1, such that there is some collection of subsets, C_2, of S_1 (the union of which is equal to S_1) such that each one of these subsets, S_2, satisfies some formula P.

The feature ⟨predicate scope⟩, and the pol and negreading features determine how the above subject and object rules are combined. However, the other thing that the scope feature determines is the structure of the *verbal rule* as shown below.

Feature : Value	Verbal Rule
⟨predicate scope⟩ : subjectwide	$\lambda S_2, S_1, V.\langle S_1, S_2 \rangle \in V$
⟨predicate scope⟩ : objectwide	$\lambda S_1, S_2, V.\langle S_1, S_2 \rangle \in V$

The relation V will be supplied by the semantic interpretation when the rule is utilized.

The subject, object and verbal rules can be combined to derive the final verbal interpretation rule. For positive polarity verbal relations (i.e., those relations in which there is no pol feature with the value $-$) the complete rule is derived as below, where I will use the abbreviations, S for subject rule, O for object rule and P for verbal rule.

Feature : Value	Complete Rule
⟨predicate scope⟩ : subjectwide	$(S(O(P)))$
⟨predicate scope⟩ : objectwide	$(O(S(P)))$

For the subjectwide scope reading above we derive the final rule by applying the verbal rule to the object rule and applying the result to the subject rule. Note that the verbal rule V will be different for each scope possibility as previously defined above. I will assume variable names in each rule can be systematically changed to avoid unwanted clashes.

For negative polarity verbal relations the possibilities are more extensive. The features ⟨subject pol⟩, ⟨object pol⟩, and ⟨predicate pol⟩ determine whether the rule constraints imposed by the features negreading apply. For negative polarity readings there are, as has been discussed, three possible negative readings: sentence negation, verb phrase negation and verb negation. Each possibility introduces negation at a different point in the complete interpretation rule. There are 27 different possible combinations of negative reading, three for each of the subject argument, object argument and verbal relation itself. The negreading feature from each argument determines whether negation is introduced at one of three possible locations within the rules. Shown below are the possibilities for subject and object scope readings. The pol and negreading features are meant to generically refer to either of the three possible combinations of pol and negreading in the subject, object and predicate complex features.

Feature : Value	Complete Rule
⟨predicate scope⟩ : subjectwide pol : $-$, negreading : s	$\neg(S(O(P)))$
⟨predicate scope⟩ : subjectwide pol : $-$, negreading : vp	$(S\neg(O(P)))$
⟨predicate scope⟩ : subjectwide pol : $-$, negreading : v	$(S(O\neg(P)))$
⟨predicate scope⟩ : objectwide pol : $-$, negreading : s	$\neg(O(S(P)))$
⟨predicate scope⟩ : objectwide pol : $-$, negreading : vp	$(O\neg(S(P)))$
⟨predicate scope⟩ : objectwide pol : $-$, negreading : v	$(O(S\neg(P)))$

A verbal predicate with a single negative polarity feature is provided with the negative reading as specified above. A predicate with two or three negative polarity features is supplied with the verbal reading specified by the combination of negreading features. That is, given a verbal predicate with two negative polarity features along with associated negreading features which both specify vp negation, the two vp negation effects cancel each other out to produce what is in effect a positive polarity reading. However, the negreading features may have different values in which case the negations will have an effect. Although there are 27 possible combinations of negative reading values from the three negreading features there are only seven different resulting readings, three of which have been shown above for subject and object scopes. I will illustrate the other four readings below for a subject wide scope reading.

-

Feature : Value	Complete Rule
⟨predicate scope⟩ : subjectwide pol : $-$, negreading : v pol : $-$, negreading : vp	$(S\neg(O\neg(P)))$
⟨predicate scope⟩ : subjectwide pol : $-$, negreading : s pol : $-$, negreading : v	$\neg(S(O\neg(P)))$
⟨predicate scope⟩ : subjectwide pol : $-$, negreading : s pol : $-$, negreading : vp	$\neg(S\neg(O(P)))$
⟨predicate scope⟩ : subjectwide pol : $-$, negreading : v pol : $-$, negreading : vp pol : $-$, negreading : s	$\neg(S\neg(O\neg(P)))$

I will look more closely at multiple negative features in the next section when the verbal analysis of monotone decreasing quantifiers is examined. I will not discuss the feature ⟨predicate aarel⟩ in this chapter as it concerns the processing of anaphor-antecedent relations which will be dealt with in detail in the next chapter.

Given an interpretation rule $\lambda C_1, C_2, V.\phi$ derived as described above we need to apply it to the particular denotation sets, $\mathcal{C}$ and $\mathcal{C}'$ for the subject and object arguments along with the relation $F(\mathsf{PRED})$, where PRED is the verbal predicate. That is we want the interpretation rule:

(224) $(((\lambda C_1, C_2, V.\phi(\mathcal{C})), (\mathcal{C}')), (F(\mathsf{PRED})))$

We can use this interpretation rule to determine a mapping from $\langle \mathcal{C}, \mathcal{C}', R\rangle$ to $\langle \mathcal{C}_s, \mathcal{C}'_s, R_s\rangle$ by taking the component-wise union of **all** the triples $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t\rangle$, where $\mathcal{C}_t \subseteq \mathcal{C}$, $\mathcal{C}'_t \subseteq \mathcal{C}'$ and $R_t \subseteq R$ such that $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t\rangle$ **minimally** satisfies the constraints in (224). A triple $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t\rangle$ minimally satisfies an interpretation rule if the deletion of any set in $\mathcal{C}_t$ or $\mathcal{C}'_t$ or any element of R_t would mean that the resulting triple no longer satisfied the interpretation rule in (224).

To illustrate this, let us take as an example the subject and object simple collective reading with positive polarity and subjectwide scope. This reading is described by the control feature structure in (225).

$$(225)\quad \text{control}\begin{bmatrix} \text{subject} & \begin{bmatrix} \text{reading} & \text{collective1} \\ \text{pol} & + \\ \text{negreading} & \text{N1} \\ \text{uniq} & - \end{bmatrix} \\ \text{object} & \begin{bmatrix} \text{reading} & \text{collective1} \\ \text{pol} & + \\ \text{negreading} & \text{N2} \\ \text{uniq} & - \end{bmatrix} \\ \text{predicate} & \begin{bmatrix} \text{pred} & \text{PRED} \\ \text{pol} & + \\ \text{scope} & \text{subjectwide} \end{bmatrix} \end{bmatrix}$$

This feature structure is given the verbal interpretation rule given below.

(226) $\lambda C_1, C_2, V.\exists S_1 \in C_1 : \exists S_2 \in C_2 : \langle S_1, S_2 \rangle \in V$

A mapping will be derived between $\langle \mathcal{C}, \mathcal{C}', R, \rangle$, and $\langle \mathcal{C}_s, \mathcal{C}'_s, R_s \rangle$, where $\mathcal{C}$ is the denotation set for the subject argument, $\mathcal{C}'$ is the denotation set for the object argument and R is the unrestricted relation between $\mathcal{C}$ and $\mathcal{C}'$. The rule in (226) can be applied to the two denotation sets describing the arguments to the verbal relation along with the value of $F(\text{PRED})$ as shown in (227), whose lambda reduced equivalent is shown in (228).

(227) $(((\lambda C_1, C_2, V.\exists S_1 \in C_1 : \exists S_2 \in C_2 : \langle S_1, S_2 \rangle \in V(\mathcal{C})), (\mathcal{C}')), (F(\text{PRED})))$

(228) $\exists S_1 \in \mathcal{C} : \exists S_2 \in \mathcal{C}' : \langle S_1, S_2 \rangle \in F(\text{PRED})$

Let us assume the following values:

- $\mathcal{C} = \{\{a,b\},\{c,d\}\}$
- $\mathcal{C}' = \{\{e,f\},\{g,h\}\}$
- $R = \{\langle\{a\},\{c\}\rangle, \langle\{a\},\{d\}\rangle, \langle\{b\},\{c\}\rangle, \langle\{b\},\{d\}\rangle, \langle\{a,b\},\{e,f\}\rangle\}$
- $F(\text{PRED}) = \{\langle\{a,b\},\{e,f\}\rangle\}$

There is only one triple $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle$ which minimally satisfies the interpretation rule in (228), as shown below.

(229) $\langle\{\{a,b\}\},\{\{e,f\}\},\{\langle\{a,b\},\{e,f\}\rangle\}\rangle$

In particular, the triple given in (230) satisfies the interpretation rule but is **not** minimal, as the deletion of the set $\{g,h\}$ would derive a triple (the triple in (229)) which still satisfies the verbal interpretation rule.

(230) $\langle\{\{a,b\}\},\{\{e,f\},\{g,h\}\},\{\langle\{a,b\},\{e,f\}\rangle\}\rangle$

The component-wise union of the satisfying triples is just the triple in (229), given that in this case there is only one such satisfying triple. The values identified by this triple can be used to describe the new vertices for the subject and object arguments and any relational edge defined between them.

I shall now provide the formal semantic interpretation of a verbal predicate, where M is a model and I is a set of identifiers.

Given a feature structure whose semantic feature sem has the value α:

$$\text{If} \begin{bmatrix} \text{control} & \begin{bmatrix} \text{subject} & \text{S} \\ \text{object} & \text{O} \\ \text{predicate} & \text{P} \end{bmatrix} \\ \text{arg1} & \text{X} \\ \text{arg2} & \text{Y} \end{bmatrix}_{tv} \sqsubseteq \alpha, \text{and} \begin{pmatrix} \text{V} = \alpha/\langle \text{control predicate pred} \rangle \\ \text{ARG1} = \alpha/\langle \text{arg1} \rangle \\ \text{ARG2} = \alpha/\langle \text{arg2} \rangle \\ \text{CTRL} = \alpha/\langle \text{control} \rangle \\ \text{ANAPHOR} = \alpha/\langle \text{control anaphor} \rangle \end{pmatrix} \text{then,}$$

$[\![\alpha]\!]^{M,I} = \langle (G_1, \mathcal{D}_1), (i_1, G_4, \mathcal{D}_3 \cup \{G_4\}) \rangle$ where

- $[\![\text{ARG1}]\!]^{M,I} = \langle (G_1, \mathcal{D}_1), (i_1, G_2, \mathcal{D}_2) \rangle$
- $[\![\text{ARG2}]\!]^{M,I} = \langle (G_2, \mathcal{D}_2), (i_2, G_3, \mathcal{D}_3) \rangle$
- $v = \langle i_1, \mathcal{C} \rangle$ where $\langle i_1, \mathcal{C} \rangle \in G_3$ and $v' = \langle i_2, \mathcal{C}' \rangle$ where $\langle i_2, \mathcal{C}' \rangle \in G_3$
  The vertices for each argument are determined via the identifiers i_1 and i_2.
- $R = \{\langle X, Y \rangle | \exists S_1 \in \mathcal{C}, \exists S_2 \in \mathcal{C}' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The unrestricted relation R allows any pair of subsets from either argument.
- If ϕ is the interpretation rule derived from CTRL then:
  - If SCOPE = subjectwide, $\phi' = (((\phi(\mathcal{C})), (\mathcal{C}')), (F(V)))$
  - If SCOPE = objectwide, $\phi' = (((\phi(\mathcal{C}')), (\mathcal{C})), (F(V)))$

  The verbal reading rule is derived, the arguments to ϕ being given in an order determined by the feature scope.
- There is a mapping from $\langle \mathcal{C}, \mathcal{C}', R \rangle$ to $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ where $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ is the component-wise union of all triples, $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle, \mathcal{C}_t \subseteq \mathcal{C}, \mathcal{C}'_t \subseteq \mathcal{C}', R_t \subseteq R$ which minimally satisfy the interpretation rule ϕ'.
  The set of triples $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle$ which satisfy the verbal reading described by ϕ' are collected together in $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$.
- $v_s = \langle i_1, \mathcal{C}_s \rangle$ and $v_o = \langle i_2, \mathcal{C}_o \rangle$.
  New vertices are constructed.
- If $R' = \{\}$ then $G_4 = cons(G_3[v_s/v, v_o/v'])$ else
  $G_4 = cons(G_3[[v_s/v, v_o/v'][\langle v_s, v_o, R' \rangle])$
  If the derived relation R' is empty no relational edge is constructed between the new vertices. The function cons conforms the new graphs to be maximally consistent.

Monotone Decreasing Quantifiers

I will now review the analysis of monotone decreasing quantifiers. Monotone decreasing quantifiers have a semantic representation of the form given below.

(231) $$\left[\begin{array}{ll} \textsf{control} & \left[\begin{array}{ll} \textsf{pred} & \textsf{PRED} \\ \textsf{number} & \textsf{NUM} \\ \textsf{uniq} & \textsf{UNIQ} \\ \textsf{pol} & - \end{array}\right] \\ \textsf{arg1} & \end{array}\right]_{det}$$

The important characteristic is that they are given negative polarity. In the semantic analysis for a monotone decreasing quantifier $Det_D A$, for model domain D and argument set A, the rule to derive the denotation set v describing the quantifier $Det_D A$ is given below.

- If $Det_D A$ is monotone decreasing then $\mathcal{C} = \{A - w | w \in Det_D A\}$

That is, the sets $A - w$ where w is a witness set for the quantifier as defined by Barwise and Cooper (1981, p. 191). For the monotone decreasing quantifiers *no farmer* and *few farmers* this produces denotations which can be *loosely* paraphrased as being similar to *every farmer* and *many farmers*, respectively. To provide the correct verbal reading we must assume that the negative polarity of these quantifiers is passed up to the control information of any verbal relation. This means that the analysis of (232) and (234) can be paraphrased as (233) and (235).

(232) No farmers own a donkey.

(233) Every farmer does not own a donkey.

(234) Some farmers own few donkeys.

(235) Some farmers do not own many donkeys.

The most appropriate negative reading seems to be that of verb phrase negation for subject monotone decreasing quantifiers and verb negation for object monotone decreasing quantifiers. That is, *no farmers own a donkey* means that *every farmer does not own any donkey*. Of course, an object wide scope reading, amongst other variations, is also possible in which *no farmer owns some particular donkey*. In the case of (235) applying verb phrase negation means that this sentence can be paraphrased as *some farmers do not own at least one set of many donkeys*. It would be a simple matter for a grammar constructing GTS semantic representations to enforce particular values for the negreading feature for negative polarity quantifiers appearing in subject and object positions.

Interesting readings occur when two monotone decreasing quantifiers are arguments to a single verbal predicate. Due to their translation as negative polarity arguments this introduces a double negative effect. However, as the proposed negative readings are different for subject and object monotone decreasing quantifiers the negations do not cancel each other out. For instance, utilizing the determiners *few* and *no* again we can produce a variety of sentences as shown below.

(236) No farmers own few donkeys.

(237) Few farmers own no donkeys.

(238) Few farmers own few donkeys.

(239) No farmers own no donkeys.

As proposed above, the subject monotone decreasing quantifiers seem to prefer verb phrase negation and the object monotone decreasing quantifiers seem to prefer verb negation. The combined effect correctly obtains the most likely reading for these sentences shown below, where (240) to (243) are the paraphrases for (236) to (239), respectively.

(240) Every farmer owns many donkeys.

(241) Many farmers own at least one donkey.

(242) Many farmers own many donkeys.

(243) Every farmer owns at least one donkey.

It is also possible for verbal negation to interact with monotone decreasing quantifiers, as shown below.

(244) No farmers do not own a donkey.

If we assume that the negative polarity verbal predicate requires verb phrase negation, this cancels out the verb phrase negation introduced by the subject monotone decreasing quantifier, producing a positive polarity sentence which can be paraphrased as below.

(245) Every farmer owns a donkey.

Again, a object wide scope reading can be given where (244) means that *every farmer owns some particular donkey*.

The GTS semantic framework gives scope for a variety of verbal readings involving monotone decreasing quantifiers. Many of these interpretations may well be unacceptable. The brief discussion above has proposed certain negative readings for subject and object monotone decreasing quantifier. The GTS semantic framework itself does not enforce a particular negative reading for monotone decreasing quantifiers. However, as will be discussed further in section 6.2 of the next chapter, the GTS framework does provide a means of stipulating constraints so that particular theories of discourse, and in particular discourse anaphora, can be derived.

5.4 Truth Determination

The truth or falsity of a particular sentence under a particular interpretation can be determined from the semantic analysis in the following manner.

- Given a sentence S whose semantic representation is the feature structure α, α is *true* with respect to a model M, set of identifiers I and discourse space $\mathcal{D}$, if $[\![\alpha]\!]^{M,I} = \langle (\langle \{\}, \{\}, \{\} \rangle, \mathcal{D}), (i, G, \mathcal{D}') \rangle$ and $\exists v \in G : v \neq \langle j, \{\} \rangle$ for some j.

The above constraint formalizes the intuition that a sentence is truthful under an interpretation if there are model-theoretic structures[3] within the model which satisfy the interpretation. That is, a semantic representation is *false* with respect to a particular interpretation if the derived graph contains vertices all of which are empty.

5.5 Compositionality

The semantic framework as outlined in the previous text allows the compositional semantic interpretation of simple discourses. However, the semantic interpretation rules provided assume semantic representations that describe fully instantiated semantic predicates. That is, for determiner predicates it is assumed a nominal phrase exists whose semantic representation is given in the argument to the determiner predicate. Similarly, for the interpretation of a verbal predicate it is assumed the semantic representation of both arguments exist. This situation, however, will not be true for certain linguistic constituents, such as the verb phrase *owns a donkey* or the determiner *every* whose semantic representations are given below.

(246) owns a donkey.

$$\begin{bmatrix} \textsf{control} & \textsf{CONTROL} \\ \textsf{arg1} & \textsf{ARG1} \\ \textsf{arg2} & \begin{bmatrix} \textsf{control} & \textsf{CONTROL2} \\ \textsf{arg1} & \begin{bmatrix} \textsf{control} & \textsf{CONTROL3} \end{bmatrix}_{n} \end{bmatrix}_{det} \end{bmatrix}_{tv}$$

(247) every

$$\begin{bmatrix} \textsf{control} & \textsf{CONTROL} \\ \textsf{arg1} & \textsf{ARG1} \end{bmatrix}_{det}$$

In both these example cases semantic representations exist but they are not fully instantiated, as arguments to the top-level predicate, tv in (246) or det in (247), will be missing. That is, in both (246) and (247) the feature arg1 has a value ARG1 which is an uninstantiated feature variable. At present, the semantic interpretation as described is simply not defined for non-fully instantiated semantic feature structures. That is, denotations can be provided for linguistic constituents which derive fully instantiated semantic representations, such as *every farmer* or *farmer who owns a donkey* but denotations can not be provided for linguistic constituents that do not derive fully instantiated semantic representations, such as *every*, *owns a donkey* or *owns*. We have to decide what an appropriate interpretation for the uninstantiated feature variable (ARG1 in the above phrases) should be if an interpretation is to be given to these types of constituents. Normally, the interpretation of a feature structure α will have the following specification.

(248) $[\![\alpha]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G', \mathcal{D}') \rangle$

An intuitive interpretation for an uninstantiated feature variable is that it describes all possible graphs and discourse spaces allowable as extensions of $\mathcal{D}$. However, as the feature α may describe a phrase of unbounded length, the number graphs that α can describe is denumerably infinite. For each such graph an interpretation can be given. Unlike the analysis of fully instantiated

[3] Sets of individuals, including the empty set.

predicates the analysis of non-fully instantiated predicates derives an infinite number of discourse spaces, $\mathcal{D}'$. Therefore, non-fully instantiated predicates can be semantically interpreted but they do not provide a unique or even finitely many extensions to a discourse. For this reason, although non-fully instantiated predicates could be given an interpretation by the framework they will be assumed not to be of interest for the analysis of pronominal noun phrase anaphora in discourse.

Chapter 6

A Semantic Framework for Anaphoric Discourse

The last chapter has described a semantic framework, GTS, which can analyse simple non-anaphoric declarative extensional discourse. This chapter, will extend GTS to handle simple forms of noun phrase anaphora in which the anaphor is a (third person) pronoun and the antecedent is derived from one or more lexical nouns or proper names introduced into the discourse.

The chapter begins with an overview of the anaphoric analysis proposed illustrating in broad detail how the different parts of the analysis tie together. Following this, three sections will provide in detail the interpretation of anaphors and their resulting anaphor-antecedent relations. I will then look at how the GTS framework could be used to impose particular theories of anaphoric constraint. I will end the chapter with a worked example followed by the investigation of particular linguistic situations and their analysis within the GTS framework.

6.1 A Brief Review of the Anaphoric Analysis

An overview of the anaphoric analysis proposed has already been given in sections 4.4.3 and 4.4.4 of chapter 4. I will briefly review the pertinent points here. Anaphors are treated separately from the resulting anaphor-antecedent relations. The treatment of anaphors extends the denotation graph built so far during an interpretation with information derived from graphs built through the previous analysis of a discourse. That is, antecedents are associated with vertices in graphs within the discourse space, the discourse space being threaded through the interpretation of a discourse. For both referential and bound anaphors information derived from vertices taken from graphs in the discourse space is used to construct a new vertex for the anaphor. However, for bound anaphors the graphs containing the antecedent vertices for the anaphor are incorporated into the graph of the anaphor; anaphoric edges are provided linking the anaphor vertex with its antecedent vertices. The treatment of anaphors will be given in section 6.1.5.

Anaphor-antecedent relations are handled within the processing of verbal relations. The denotational arguments to a transitive verbal relation are two vertices from a denotation graph. Each vertex will describe a denotation set, the sets from which will be utilized in satisfying the particular verbal reading applied to the verbal relation under the interpretation. The *weak* anaphor-antecedent relation is enforced by limiting which sets from the argument denotation sets can be

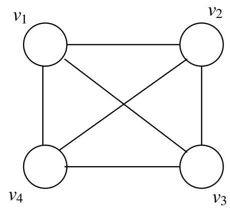


Figure 6.1: A graph.

used in the satisfaction of the verbal reading applied. This is accomplished by treating the denotation graph involved in the verbal relation as a constraint network to which a consistent labelling must be found. The *strong* anaphor-antecedent relation can be determined by an appropriate constraint relating the sets from the denotation set arguments before the verbal reading with those sets from the denotation set arguments which satisfy the verbal reading. The particular constraint utilized to determine a strong anaphor-antecedent relation will be discussed in section 6.1.6, along with the details concerning the imposition of a weak anaphor-antecedent relation.

In section 4.4.4 of chapter 4 the treatment of denotation graphs as constraint networks was discussed. This involved finding a label for each vertex of a denotation graph such that the relational and anaphoric edge constraints were satisfied. However, only a preliminary anaphoric edge constraint was provided which dealt only with anaphors which refer to single antecedents. In section 6.1.1, this anaphoric edge constraint is reformulated to cope with anaphors which reference multiple antecedents.

The interpretation function, $[\![]\!]$, will need to be extended to cope with anaphoric discourse. The extensions to the interpretation function are discussed in section 6.1.2. The (minimally) revised interpretations for lexical noun predicates and generalized quantifiers are provided in sections 6.1.3 and 6.1.4 respectively.

6.1.1 Consistent Labelling of a Graph

In section 5.2 of the last chapter, I discussed how a graph could be treated as a constraint network and consistently labelled so as to respect the relational and anaphoric constraints. However, the anaphoric edge labelling constraint provided assumes that any vertex has at most a single anaphoric edge associated with it. This will be false in situations where anaphors refer to multiple antecedents. In this case, the vertex associated with an anaphor will have multiple anaphoric edges emanating from it. To handle this more complex scenario, the notion of an anaphoric circuit will be defined. A *walk* through a graph in standard graph theory is a list of vertices such that adjacent vertices in the list are connected by an edge. A *circuit* is a walk through a graph which begins and ends at the same vertex. A *simple* circuit is a circuit in which any vertex on the circuit appears only once on the circuit, excluding the start and end vertices which must be the same. Given the graph in figure 6.1, some possible (simple) circuits are, v_1, v_2, v_3, v_4, v_1 and v_1, v_3, v_2, v_1.

The type of circuit utilized by GTS will be called an *anaphoric circuit* and will be constrained by the notion of an *anaphor edge set*. Given a graph $G = \langle V, E, A \rangle$ and $v \in V$ then the anaphor edge set for v will be the set of anaphoric edges $\langle v, v' \rangle \in A$ for some v'. For anaphoric references to

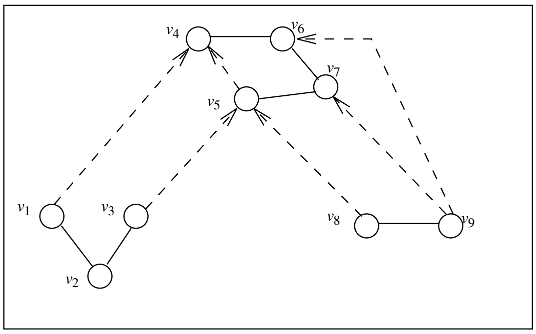


Figure 6.2: An example denotation graph

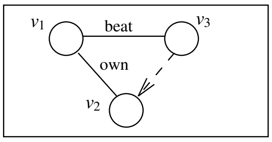


Figure 6.3: The graph structure for the standard quantified donkey sentence.

single antecedents the anaphor edge sets for these anaphors will only contain a single anaphoric edge. The concept of an *anaphoric* circuit can now be defined.

Definition 20: An *anaphoric* circuit through a denotation graph G is a *simple* circuit passing through vertices connected by relational or anaphoric edges but which passes through at most a single anaphoric edge from each anaphoric edge set in G.

I shall call the anaphoric edges which appear on anaphoric circuits, *activated* edges and the antecedents identified by an anaphoric circuit *activated* antecedents. Some example anaphoric circuits can be illustrated by looking at the denotation graph in figure 6.2, for example $\mathcal{A}_1 = v_3, v_5, v_8, v_9, v_6, v_4, v_1, v_2, v_3$ and $\mathcal{A}_2 = v_8, v_5, v_7, v_9, v_8$. However, the following circuit is not an *anaphoric* circuit, v_9, v_7, v_6, v_9, as it traverses anaphoric edges from the anaphoric edge set for v_9 twice, once from v_9 to v_7 and again from v_6 to v_9 [1].

In simple graphs such as those describing the standard quantified donkey sentence there may be only one anaphoric circuit. This can be seen by repeating the graph that describes the standard quantified donkey sentence (*every farmer who owns a donkey beats it*) within a satisfying model. This graph is shown in figure 6.3. The only anaphoric circuit in this graph is v_1, v_2, v_3. In more complex situations this may not be the case and there may be several anaphoric circuits.

Anaphoric circuits will be used to redefine the anaphoric edge labelling constraint to correctly handle anaphors which refer to multiple antecedents. In order to accomplish this I will need to define when two anaphoric circuits are in *conflict*.

Definition 21: Two anaphoric circuits for a graph G are in conflict if they utilize distinct anaphoric edges from any anaphoric edge set for a vertex in G.

[1] Most nontrivial anaphoric circuits will have many descriptions as lists of vertices. That is, the anaphoric circuit $\mathcal{A}_2$ above could be identified as, v_5, v_7, v_9, v_8, v_5. However, I will assume, following standard graph theory, that the different descriptions identify the same underlying circuit.

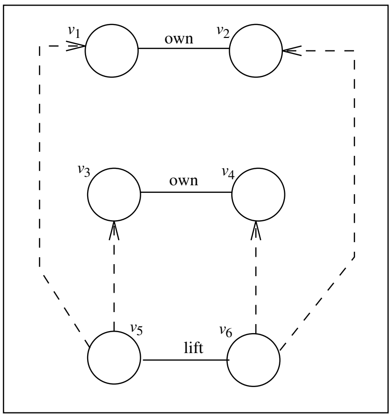


Figure 6.4: The graph describing the discourse in (249).

As an example, the previous anaphoric circuits $\mathcal{A}_1$ and $\mathcal{A}_2$ for the graph in figure 6.2 are in conflict as they together utilize two distinct anaphoric edges emanating from the vertex v_9. I shall now extend the above definition to a set of anaphoric circuits.

Definition 22: A set of anaphoric circuits is in conflict if at least two circuits within that set are in conflict.

A set of non-conflicting anaphoric circuits for a graph G is *maximal* if the addition of any other anaphoric circuit for G would cause the circuits to be in conflict. Utilizing these concepts, the revised anaphoric edge labelling constraint will be as below, with respect to a graph G.

- Anaphoric Edge Constraint.
  Let $\mathcal{M}$ be the set of sets of maximal non-conflicting anaphoric circuits for G. If $\mathcal{M}$ is not empty then there must be some set of circuits $m \in \mathcal{M}$ such that the label on each anaphor vertex identified in m is identical to the label of its activated antecedent identified in m.

The constraint essentially ensures that each anaphor vertex must be given the same label as one of its antecedent vertices. However, the choice of antecedents is constrained globally by the anaphoric circuits in the graph in question.

The example below will help illustrate the process.

(249) John owns a table. Mary owns a table. They lift them.

Assuming bound anaphoric references, the graph derived for the entire discourse in a satisfying model is shown in figure 6.4. The first sentence in (249) would have derived the graph segment with the vertices v_1 and v_2: the vertex v_1 would contain the individual John and the vertex v_2 would contain the tables he owns. Similarly, the second sentence would have derived the graph segment with the vertices v_3 and v_4: the vertex v_3 would contain the individual Mary and the vertex v_4 would contain the tables she owns. There are two anaphoric circuits in this graph, $\mathcal{A}_1 = v_5, v_1, v_2, v_6, v_5$ and $\mathcal{A}_2 = v_5, v_3, v_4, v_6, v_5$. The two circuits are in conflict with each other. Therefore, the set $\mathcal{M}$ of maximal non-conflicting circuits will be simply $\{\{\mathcal{A}_1\}, \{\mathcal{A}_2\}\}$. I am assuming that the anaphor *they* is assigned the vertex v_5 and contains the set of John and Mary, and the anaphor *them* is assigned the vertex v_6 and contains the sets of single tables that John and Mary own. If we treat this graph as a constraint network and attempt to find a consistent labelling then we must choose one of the two sets $\{\mathcal{A}_1\}$ and $\{\mathcal{A}_2\}$ in order to satisfy the anaphoric labelling constraint. In the present example, where we have labeled v_5 with the set containing the individual John, utilizing the set containing only the anaphoric circuit $\mathcal{A}_1$ is appropriate. In that case the anaphor vertex v_5 has one antecedent on the circuit $\mathcal{A}_1$, in particular v_1. The only available label for v_1 will be the set containing John and this is the same as that given to v_5 which satisfies the anaphoric edge labelling constraint. The case for the vertex v_6 is similar. The only antecedent on a chosen anaphoric circuit is v_2. If the label on v_2 is the same as the label on v_6 the anaphoric edge labelling constraint will be satisfied. If we had been labelling v_5 with the set containing Mary the set containing the anaphoric circuit $\mathcal{A}_2$ would be appropriate.

6.1.2 The Interpretation Function

The interpretation function will need to be extended to deal with anaphoric discourses. I will begin by summarizing the basics of the interpretation of discourses detailed in the last chapter. A discourse is a temporally ordered set of declarative extensional sentences $S_1, S_2, ..., S_n$. An appropriate (compositional) analysis will, given a sentence in the discourse, S_j, derive an appropriate feature matrix A_j representing the syntactic and semantic information of S_j. The value of the sem feature of A_j will be α_j, which will contain the semantic representation derived from S_j. That is, the analysis of a discourse $S_1, S_2, ..., S_n$ will produce a series of semantic representations $\alpha_1, \alpha_2, ..., \alpha_n$. The semantic interpretation function is applied to each semantic representation α_j, using the following inductive procedure, where M is a model and I is a set of identifiers.

(250) $[\![\alpha_1]\!]^{M,I} = \langle((\langle\{\},\{\},\{\}\rangle,\{\}),(i,G_1,\mathcal{D}_1)\rangle$

(251) $[\![\alpha_j]\!]^{M,I} = \langle((\langle\{\},\{\},\{\}\rangle,\mathcal{D}_{j-1}),(i,G_j,\mathcal{D}_j)\rangle$

For the interpretation of anaphoric discourse, however, I shall define two functions, an anaphoric constraint function $\mathcal{CONS}$ and an anaphoric resolution function $\mathcal{RES}$ with respect to which the interpretation will be given. These functions will play a central role in the interpretation of anaphors. The specification of the interpretation function is now as shown below where α is a semantic representation, G and G' are graphs, and $\mathcal{D}$ and $\mathcal{D}'$ are discourse spaces and i is an identifier.

(252) $[\![\alpha_p]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$

That is, the interpretation of discourses will now be carried out using the following inductive procedure.

(253) $[\![\alpha_1]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle((\langle\{\},\{\},\{\}\rangle,\{\}),(i,G_1,\mathcal{D}_1)\rangle$

(254) $[\![\alpha_j]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle((\langle\{\},\{\},\{\}\rangle,\mathcal{D}_{j-1}),(i,G_j,\mathcal{D}_j)\rangle$

The truth or falsity of a particular sentence under the revised interpretation can be determined from the semantic analysis in the following manner.

- Given a sentence S whose semantic representation is the feature structure α, α is *true* with respect to a model M, set of identifiers I, a semantic interpretation function $\mathcal{CONS}$, a semantic resolution function $\mathcal{RES}$, and discourse space $\mathcal{D}$, if $[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(\langle\{\},\{\},\{\}\rangle,\mathcal{D}),(i,G,\mathcal{D}')\rangle$ and $\exists v \in G : v \neq \langle j,\{\}\rangle$ for some j.

The anaphoric constraint function $\mathcal{CONS}$ will determine, given a semantic representation for an anaphor, the current discourse space and a graph from that discourse space, a set of acceptable denotations for the anaphor and its antecedents. The specification of this function is given below.

(255) An *anaphoric constraint function* $\mathcal{CONS}$ takes a triple $\langle\alpha,\mathcal{D},G\rangle$ where α is the semantic representation for an anaphor, $\mathcal{D}$ is a discourse space and $G \in \mathcal{D}$ is a denotation graph and returns a set of *anaphor antecedent denotation pairs* $\langle\mathcal{C},\mathcal{R}\rangle$ where $\mathcal{C}$ is a denotation set (for an anaphor) and is $\mathcal{R}$ a set of vertex graph pairs (for the antecedents) $\langle\langle i,\mathcal{C}'\rangle,G'\rangle$ such that $G' \in \mathcal{D}$, $\langle i,\mathcal{C}'\rangle \in G'$ and every individual in $\mathcal{C}'$ is in $\mathcal{C}$.

The function $\mathcal{CONS}$ is specified only very broadly with only some basic constraints relating its arguments with the result returned. No explicit indication is given as to how particular $\mathcal{CONS}$ functions can be defined in detail. The GTS framework only requires that some function be defined which validates the specification above. An example $\mathcal{CONS}$ function covering semantic number constraints will be discussed in section 6.2.4.

The anaphoric resolution function $\mathcal{RES}$ will take a *discourse context* and a set of anaphor antecedent denotation pairs $\mathcal{P}$ and returns one pair from $\mathcal{P}$ describing the denotation set to be provided for the anaphor and the vertex-graph pairs describing the antecedents of the anaphor. Within the interpretation rules the $\mathcal{RES}$ function will be applied to the output of the $\mathcal{CONS}$ function. That is, the $\mathcal{CONS}$ function will specify the acceptable anaphor antecedent pairs and the $\mathcal{RES}$ function will choose one pair from the available selection. This thesis is not concerned with anaphoric resolution. For this reason I will leave unspecified the nature of the *discourse context* required by the anaphoric resolution function. I will assume that the discourse context may hold what ever information is required by the anaphoric resolution function. The specification of the function $\mathcal{RES}$ is given below.

Definition 23: An *anaphoric resolution function* $\mathcal{RES}$ takes a discourse context and a set $\mathcal{P}$ of anaphor antecedent denotation pairs and returns a pair $\langle\mathcal{C},\mathcal{R}\rangle \in \mathcal{P}$ where $\mathcal{C}$ is a denotation set and $\mathcal{R}$ is a set of vertex graph pairs or returns $\langle\{\},\{\}\rangle$ if $\mathcal{P} = \{\}$.

Thus the function $\mathcal{RES}$ is required to uniquely determine for any discourse context an anaphor denotation set and antecedent denotation from the set of acceptable pairs $\mathcal{P}$ or return a null answer if there are no such pairs in $\mathcal{P}$.

For completeness, I shall provide the interpretations of lexical nouns and generalized quantifiers from the previous chapter which differ **only** with respect to the revised nature of the interpretation function.

6.1.3 The Interpretation of Lexical Nouns

The semantic interpretation for a lexical noun predicate is given below, where M is a model, I is a set of identifiers, $\mathcal{CONS}$ is an anaphoric constraint function, $\mathcal{RES}$ is an anaphoric resolution function and α is a feature-based semantic representation.

If $\begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{A} \\ \textsf{number} & \textsf{B} \end{bmatrix} \end{bmatrix}_n \sqsubseteq \alpha$ and $\begin{pmatrix} \textsf{PRED} = \alpha/\langle\textsf{control pred}\rangle \\ \textsf{NUM} = \alpha/\langle\textsf{control number}\rangle \end{pmatrix}$ then
$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G[v],\mathcal{D}\cup\{G[v]\})\rangle$, where,

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$.
- $v = \langle i, \mathcal{C}\rangle$ where $\mathcal{C} = \begin{cases} \{X \subseteq F(\textsf{PRED}) \| X| = 1\} & \text{If NUM} = \textsf{singular} \\ \{X \subseteq F(\textsf{PRED}) \| X| \geq 1\} & \text{If NUM} = \textsf{plural} \end{cases}$

6.1.4 The Interpretation of Generalized Quantifiers

The semantic interpretation of a determiner predicate is given below, where M is a model, I is a set of identifiers $\mathcal{CONS}$ is an anaphoric constraint function, $\mathcal{RES}$ is an anaphoric resolution function and α is a feature-based semantic representation.

If $\begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{A} \\ \textsf{number} & \textsf{B} \\ \textsf{uniq} & \textsf{C} \\ \textsf{pol} & \textsf{D} \end{bmatrix} \\ \textsf{arg1} & \textsf{G} \end{bmatrix}_{det} \sqsubseteq \alpha$ and $\begin{pmatrix} \textsf{PRED} = \alpha/\langle\textsf{control pred}\rangle \\ \textsf{NUM} = \alpha/\langle\textsf{control number}\rangle \\ \textsf{ARG} = \alpha/\langle\textsf{arg1}\rangle \end{pmatrix}$ then
$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G'',\mathcal{D}'\cup\{G''\})\rangle$ where

- $[\![\textsf{ARG}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$
- $\langle i,\mathcal{C}\rangle \in G'$, $S = \bigcup_{X\in\mathcal{C}} X$

  The vertex holding the information of the argument to the determiner is $\langle i,\mathcal{C}\rangle$. We take the union of the sets in $\mathcal{C}$.
- $\mathcal{C}' = \begin{cases} \{X \subseteq S | X = S\} & \text{If PRED} = \textsf{every} \\ \{X \subseteq S \| X| \geq \frac{1}{2}|S|\} & \text{If PRED} = \textsf{most} \\ \{X \subseteq S \| X| = 1\} & \text{If PRED} = \textsf{a} \\ \{X \subseteq S \| X| = 1\} & \text{If PRED} = \textsf{some}, \textsf{NUM} = \textsf{singular} \\ \{X \subseteq S \| X| > 1\} & \text{If PRED} = \textsf{some}, \textsf{NUM} = \textsf{plural} \\ \{X \subseteq S | X = S\} & \text{If PRED} = \textsf{no} \\ \{X \subseteq S \| X| \geq \frac{1}{2}|S|\} & \text{If PRED} = \textsf{few} \\ \{X \subseteq S \| X| = 2\} & \text{If PRED} = \textsf{two} \end{cases}$

  From the set S we can determine the denotation set $\mathcal{C}'$ containing the witness sets for the particular quantifier in question.
- $G'' = G'[\langle i,\mathcal{C}'\rangle/\langle i,\mathcal{C}\rangle]$

  The graph G'' is the graph G' with the vertex $\langle i,\mathcal{C}\rangle$ replaced by the vertex $\langle i,\mathcal{C}'\rangle$.

6.1.5 The Interpretation of Anaphors

Pronouns within a discourse will be provided with the following type of semantic representation.

$$\left[\begin{array}{ll}\textsf{control} & \left[\begin{array}{ll}\textsf{pred} & \textsf{PRO}\\ \textsf{anaphor} & \textsf{TYPE}\\ \textsf{number} & \textsf{NUM}\end{array}\right]\end{array}\right]_{pro}$$

The control feature contains three features. The feature pred specifies the particular pronominal predicate, e.g., he,she,them. The anaphor feature specifies the type of pronoun, either bound or referential. The number feature determines whether the pronoun is syntactically singular or plural.

The interpretation of bound and referential pronouns differs. Bound pronouns require that we incorporate the denotation graphs containing the antecedents to the anaphor into the denotation graph of the anaphor and create anaphoric edges from the anaphor vertex to the antecedent vertices. This is because we need to keep the constraint information of the antecedents for a bound anaphor so that bound anaphor-antecedent relations can be handled correctly. Referential pronouns simply derive a new vertex within the denotation graph being extended. The graph denotations of the antecedents are not incorporated. This is because the interpretation of referential pronouns does not depend on the constraints imposed on their antecedents.

I will assume the additional notational convention.

- If $\mathcal{R}$ is a set of vertex-graph pairs, then $\mathcal{R}_G$ is the set of graphs from $\mathcal{R}$.

The formal interpretation of referential and bound anaphors can now be directly given, where M is a model, I is a set of identifiers, $\mathcal{CONS}$ is an anaphoric constraint function, $\mathcal{RES}$ is an anaphoric resolution function, $\mathcal{DIS}$ is a discourse context and α is a feature-based semantic representation from $\alpha_1, \alpha_2, ...\alpha_n$.

$$\text{If } \left[\begin{array}{ll}\textsf{control} & \left[\begin{array}{ll}\textsf{pred} & \textsf{PRO}\\ \textsf{anaphor} & \textsf{referential}\\ \textsf{number} & \textsf{NUM}\end{array}\right]\end{array}\right]_{pro} \sqsubseteq \alpha \text{ then}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G, \mathcal{D}), (i, G', \mathcal{D} \cup \{G'\}) \rangle$ where

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$
- $\mathcal{RES}(\mathcal{DIS}, \mathcal{CONS}(\alpha, \mathcal{D}, G)) = \langle \mathcal{C}, \mathcal{R} \rangle$

  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function $\mathcal{RES}$ to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{CONS}$.
- $G' = G[\langle i, \mathcal{C} \rangle]$

  Create the graph for the anaphor by extending G with the anaphor vertex $\langle i, \mathcal{C} \rangle$.

$$\text{If } \left[\begin{array}{ll}\textsf{control} & \left[\begin{array}{ll}\textsf{pred} & \textsf{PRO}\\ \textsf{anaphor} & \textsf{bound}\\ \textsf{number} & \textsf{NUM}\end{array}\right]\end{array}\right]_{pro} \sqsubseteq \alpha \text{ then}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G',\mathcal{D}\cup\{G'\})\rangle$ where

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$
- $\mathcal{RES}(\mathcal{DIS},\mathcal{CONS}(\alpha,\mathcal{D},G)) = \langle \mathcal{C},\mathcal{R}\rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function $\mathcal{RES}$ to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{CONS}$.
- $v = \langle i,\mathcal{C}\rangle$.
  The vertex for the anaphor is created.
- $A = \{\langle v,v_i\rangle | \langle v_i,G\rangle \in \mathcal{R}\}$ and $G_{t1} = \langle\{\},\{\},A\rangle$
  A is the set of anaphoric edges linking anaphor to antecedent, and a graph G_{t1} is created to hold these anaphoric edges.
- $G_{t2} = \bigcup_{G\in\mathcal{R}_G} G$
  A graph G_{t2} is created from the union of the antecedent graphs.
- $G' = G[v] \cup G_{t1} \cup G_{t2}$
  The graph for the anaphor is the union of the extension of the graph G with the anaphor vertex along with the graphs G_{t1} and G_{t2}.

6.1.6 The Interpretation of Anaphor-Antecedent Relations and Verbal Relations

In section 4.4.3 of chapter 4, the analysis of anaphor-antecedents was outlined. It was proposed that the correct place to treat anaphor-antecedent relations was during the analysis of verbal predicates. That is, in handling the donkey sentence in (256) the crucial issue is how the beat verbal predicate is treated.

(256) Every farmer who owns a donkey beats it.

In the analysis of the beat relation it is important to ensure that we check that farmers only beat donkeys they own, thus satisfying the weak anaphor-antecedent relation. Secondly, if a strong anaphor-antecedent relation is to be enforced we must in addition ensure that if a farmer beats a donkey he owns he beats all the donkeys he owns.

The previous chapter has provided a semantics for (transitive) verbal predicates in referential situations. To treat anaphor-antecedent relations we will need to extend the interpretation given there so that it correctly handles verbal predicates in anaphoric situations while still providing the same interpretation to verbal predicates in referential situations.

The central theme of the previous chapter's treatment of verbal predicates was the definition of a mapping between $\langle \mathcal{C},\mathcal{C}',R\rangle$ and $\langle \mathcal{C}_s,\mathcal{C}_o,R'\rangle$ where $\langle i,\mathcal{C}\rangle$ and $\langle j,\mathcal{C}'\rangle$ are the vertices for the subject and object arguments to the verbal relation and R is defined as below.

- $R = \{\langle X,Y\rangle | \exists S_1 \in \mathcal{C}, \exists S_2 \in \mathcal{C}' : X \subseteq S_1 \wedge Y \subseteq S_2\}$

The resulting triple $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ is derived by determining the interpretation rule, ϕ, described by the control information of the verbal predicate and taking the component-wise union of **all** triples $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle, \mathcal{C}_t \subseteq \mathcal{C}, \mathcal{C}'_t \subseteq \mathcal{C}', R_t \subseteq R$ which **minimally** satisfy $(((\phi, (\mathcal{C})), (\mathcal{C}')), (F(\mathsf{PRED})))$, where the verbal predicate for the verbal relation is PRED, i.e., the interpretation rule ϕ applied to the subject and object denotation sets along with the value supplied by model to the verbal predicate PRED[2].

For anaphoric situations, however, we must restrict the relation R from all possible pairs of sets from $\mathcal{C}$ and $\mathcal{C}'$ to only those which are anaphorically acceptable. We can determine whether a certain pair of sets is anaphorically acceptable by treating the denotation graph from which the verbal arguments were taken as a constraint network and attempting to find a consistent labelling which verifies the pair of sets from $\mathcal{C}$ and $\mathcal{C}'$. The new restricted relation, R_a, is derived from R as below, where $v = \langle i, \mathcal{C} \rangle$ and $v' = \langle j, \mathcal{C}' \rangle$ are the argument vertices to the verbal relation and G is the denotation graph from which these vertices were taken.

- $R_a = \{\langle X, Y \rangle \in R | satis(G[\langle v, v', R \rangle], \mathcal{L}) \wedge \{\langle v, X \rangle, \langle v', Y \rangle\} \subseteq \mathcal{L}\}$

The graph G is extended with a relational edge $\langle v, v', R \rangle$, before determining a consistent labelling through the relation *satis*; the relation *satis* was defined on page 86. This extension of G allows the correct detection of anaphoric circuits without restricting the possible consistent labellings: the relation R allowing all possible pairs from v and v'.

We can view the semantic analysis of verbal relations within the anaphoric domain in the following manner. We provide an edge ($\langle v, v', R \rangle$ above) connecting the two vertices over which the verbal relation is being determined. This edge is completely general allowing every combination of sets from each argument. Firstly, through the use of *satis* and then afterwards via the imposition of a particular verbal reading we limit the valid pairs of sets from each argument specified by this edge until we are left at the end of this process with just those sets which satisfy the verbal relation, possibly none.

I shall now provide the formal semantic interpretation of a verbal predicate extended to handle anaphoric situations, where M is a model and I is a set of identifiers, $\mathcal{CONS}$ is an anaphoric constraint function and $\mathcal{RES}$ is an anaphoric resolution function and α is a feature-based semantic representation.

$$\text{If} \begin{bmatrix} \text{control} & \begin{bmatrix} \text{subject} & \text{S} \\ \text{object} & \text{O} \\ \text{predicate} & \text{P} \end{bmatrix} \\ \text{arg1} & \text{X} \\ \text{arg2} & \text{Y} \end{bmatrix}_{tv} \sqsubseteq \alpha, \text{ and } \begin{pmatrix} \text{V} = \alpha/\langle \text{control predicate pred} \rangle \\ \text{ARG1} = \alpha/\langle \text{arg1} \rangle \\ \text{ARG2} = \alpha/\langle \text{arg2} \rangle \\ \text{CTRL} = \alpha/\langle \text{control} \rangle \\ \text{AAREL} = \alpha/\langle \text{control aarel} \rangle \end{pmatrix} \text{ then,}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1, \mathcal{D}_1), (i_1, G_4, \mathcal{D}_3 \cup \{G_4\}) \rangle$ where

- $[\![\text{ARG1}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1, \mathcal{D}_1), (i_1, G_2, \mathcal{D}_2) \rangle$
- $[\![\text{ARG2}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_2, \mathcal{D}_2), (i_2, G_3, \mathcal{D}_3) \rangle$
- $v = \langle i_1, \mathcal{C} \rangle$ where $\langle i_1, \mathcal{C} \rangle \in G_3$ and $v' = \langle i_2, \mathcal{C}' \rangle$ where $\langle i_2, \mathcal{C}' \rangle \in G_3$
  The vertices for each argument are determined via the identifiers i_1 and i_2.

[2]The arguments $\mathcal{C}$ and $\mathcal{C}'$ will need to be supplied in an order dependent on the desired scope.

- $R = \{\langle X,Y\rangle | \exists S_1 \in \mathcal{C}, \exists S_2 \in \mathcal{C}' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The relation R allows any pair of subsets from either argument.

- $R_a = \{\langle X,Y\rangle \in R | satis(G_3[\langle v,v',R\rangle], \mathcal{L}) \wedge \{\langle v,X\rangle, \langle v',Y\rangle\} \subseteq \mathcal{L}\}$
  The relation R_a limits the relation R by allowing only anaphorically acceptable pairs from R. This is determined via the relation satis (defined on page 86) over the graph G extended with an edge between the vertices v and v′ utilizing the relation R. The sets X and Y are labels for the vertices v and v′ respectively.

- If ϕ is the interpretation rule derived from CTRL then:

  - If SCOPE = subjectwide, $\phi' = (((\phi(\mathcal{C})),(\mathcal{C}')),(F(V)))$
  - If SCOPE = objectwide, $\phi' = (((\phi(\mathcal{C}')),(\mathcal{C})),(F(V)))$

  The verbal reading rule is derived, the arguments to ϕ being given in an order determined by the feature scope.

- There is a mapping from $\langle \mathcal{C},\mathcal{C}',R_a\rangle$ to $\langle \mathcal{C}_s,\mathcal{C}_o,R'\rangle$ where $\langle \mathcal{C}_s,\mathcal{C}_o,R'\rangle$ is the componentwise union of all triples, $\langle \mathcal{C}_t,\mathcal{C}_t',R_t\rangle, \mathcal{C}_t \subseteq \mathcal{C}, \mathcal{C}_t' \subseteq \mathcal{C}', R_t \subseteq R_a$ which minimally satisfy the interpretation rule ϕ'
  The set of triples $\langle \mathcal{C}_t,\mathcal{C}_t',R_t\rangle$ which satisfy the verbal reading described by ϕ' are collected together in $\langle \mathcal{C}_s,\mathcal{C}_o,R'\rangle$.

- $v_s = \langle i_1,\mathcal{C}_s\rangle$ and $v_o = \langle i_2,\mathcal{C}_o\rangle$.
  New vertices are constructed.

- If $R' = \{\}$ then $G_4 = cons(G_3[v_s/v, v_o/v'])$ else
  $G_4 = cons(G_3[[v_s/v, v_o/v'][\langle v_s,v_o,R'\rangle]])$
  If the derived relation R′ is empty no relational edge is constructed between the new vertices. The function cons (defined on page 86) forces the new graphs to be maximally consistent.

For the strong anaphor-antecedent relation to hold for the example donkey sentence (256) we must check that for every farmer who owns and beats a donkey that farmer beats every donkey he owns. This can be accomplished, in general, by checking whether the relation R' has collected all possible anaphorically related pairs with respect to the subject argument, where the set of all possible anaphorically related pairs is given by R'. An appropriate realisation of this constraint is shown below.

- If AAREL = strong then $R' = \{\langle X,Y\rangle \in R_a | \exists Z : \langle X,Z\rangle \in R'\}$

To incorporate this extra rule into the interpretation, we would have to require that if the constraint was not satisfied then the two argument vertices v_s and v_o should be forced to be empty along with the relation R', thus ensuring that no sets satisfied the verbal relation.

6.2 Deriving Empirical Theories of Anaphora using GTS

In the following sections I will discuss how the GTS framework can be utilized to derive empirical theories of anaphora. The GTS framework makes available various constraint mechanisms within the representational and denotational domains. By specifying a particular set of constraints the GTS framework can be "parameterized" to derive a particular theory of anaphoric reference.

In section 6.2.1, I will discuss the general stance taken to anaphoric constraints in GTS. In section 6.2.2, I shall list the various "parameters" the framework makes available in order that an empirically sensitive theory of discourse anaphora can be derived from the semantic framework. In section 6.2.3, I will discuss a particular constraint theory imposed on the representational structures, while in section 6.2.4, I will look at a constraint theory based upon the denotational structures.

6.2.1 Anaphoric Constraints

The GTS framework is a general framework for anaphoric analysis. Specifying in a precise manner the processing of discourses with anaphors and allowing a wide range of interpretations of those discourses. However, the framework does not determine which interpretations are valid interpretations for the subset of English anaphoric discourse covered. It is the intention of the following sections to show how via the imposition of representational and denotational constraints a particular theory of anaphoric reference can be derived from the framework. The distinction I am trying draw out here is that a semantic framework which can handle the analysis of anaphoric discourse can be divorced from the application of a particular theory of anaphoric reference. GTS should be considered a *framework* onto which particular theories of anaphoric reference can be imposed.

The utility in the philosophy I am proposing is that the resulting framework may have a longer shelf life than a theory which imposes a particular viewpoint on anaphoric constraint. Furthermore, such a framework should not be caught out by new empirical data suggesting new nuances on the constraining of anaphora. However, against this, the framework must show that the mechanisms provided to express anaphoric constraints can indeed be applied to provide adequately constrained theories of anaphora.

There are two areas within GTS onto which anaphoric constraints can be enforced.

1. The feature-based representations.
2. The semantic interpretation functions $\mathcal{CONS}$ and $\mathcal{RES}$.

Unification feature grammars have had considerable success in providing powerful descriptions of natural language syntax. It would seem fruitful to apply the same feature-based techniques to semantic processing as well. However, before continuing it should be pointed out that GTS does not propose to determine how feature structures are manipulated within a particular grammar. GTS does, though, determine the minimal structure of the complex feature sem which holds the semantic representation for each linguistic constituent. That is, GTS is a semantic framework not a complete theory of grammar. Given this understanding though, the previous discussion has already highlighted several areas in which, using some standard properties common to most feature-based grammars, the semantic interpretation can be constrained.

An almost uniformly observed property found within feature-based grammatical formalisms is the ability to enforce identical feature values in two feature structures within a grammar rule, e.g., to perform unification. This ability would allow a grammar to capture some of the constraints discussed in the previous text. For example, it was noted that some quantifiers, dependent on the given determiner, require that a uniqueness constraint be applied to the analysis of their verbal relation. For instance, the determiner *three* interpreted as *exactly three* requires such a constraint. An example PATR rule, shown below, for simple verb phrases could help capture this constraint.

```
VP -> V NP:
    <VP sem control object uniq> = <NP sem control uniq>
```

By ensuring that all determiners are marked for uniqueness, and applying a similar constraint for subject noun phrases the appropriate values for the uniq features within a transitive verbal relation can be provided.

Another constraint possibility is to allow noun phrases to enforce a particular type of verbal reading. For instance, it is generally considered that noun phrases containing the head determiner *every* do not allow collective verbal readings[3]. A feature read could be provided and set for determiners that enforce a particular reading. As a final example, Kanazawa (1994), provides evidence that the monotonicity of subject noun phrase determiners in donkey sentences promotes either a weak or strong anaphor-antecedent relation. A monotonicity feature could be provided for noun phrases and passed up to the verbal relation to help choose in cases of bound anaphor-antecedent relations whether to provide a weak or strong anaphor-antecedent relation. This constraint will be implemented in detail in section 6.2.3.

The other possible location for constraints, noted above, was the semantic interpretations functions, $\mathcal{CONS}$, the anaphoric constraint function, and $\mathcal{RES}$, the anaphoric resolution function. The function $\mathcal{RES}$ determines for any discourse context and a set of pairs of anaphor antecedent denotations a pair of denotations for the anaphor and its antecedents. This thesis does not concern itself with anaphoric resolution and thus I will not discuss particular $\mathcal{RES}$ functions that might be derived, see for example, Grosz (1986), Sidner (1983), Hobbs (1986) for some theoretical discussions concerning anaphoric resolution.

The function $\mathcal{CONS}$ determines the available anaphor antecedent denotations given some anaphoric semantic representation, a discourse space and a graph from the discourse space. Denotation sets (i.e., the contents of vertices of the denotation graphs) lend themselves to semantic number constraints, a form of constraint advocated by Elworthy (1993). Antecedents, as have been discussed, are identified by vertices in some graph within a discourse space. For singular and plural referential pronouns we can require that the antecedent vertex contain a single individual or more than one individual, respectively, before accepting the reference. For bound pronouns, things are more complex as shown by the standard donkey sentence.

(257) Every farmer who owns a donkey beats it.

The anaphor is able to refer to a vertex describing the donkeys that are owned by every farmer: a vertex which may contain more than one individual (donkey). This vertex will describe a denotation set whose contents will be one or more sets of single individuals (donkeys). That is,

[3]For some discussion of this, see van der Does (1991, p. 23).

for bound anaphoric pronouns the options are slightly greater than for referential pronouns. The implementation of semantic number agreement between anaphor and antecedents will be looked at in greater detail in section 6.2.4 where a $\mathcal{CONS}$ function implementing this type of constraint will be illustrated.

Constraints might also utilize the discourse space and vertex set of a graph. Intentionally, these structures have been made as simple as possible by defining them as sets, i.e., a set of vertices for the vertex set (in a graph) and a set of denotation graphs for the discourse space. However, a particular anaphoric theory implemented within the GTS framework might wish to redefine the operations used on vertex sets and discourse spaces to allow the construction of a more complex denotational space. For instance, following the lead of DRT one could impose a structure to the discourse space and vertex sets following the hierarchical structure found in DRSs. This structure could be utilized when accepting anaphoric references in a manner similar to that defined by Kamp's accessibility relation. Alternatively, the discourse space could be divided into discourse segments in a manner described by Grosz (Grosz & Sidner, 1986), placing the denotation graphs derived from the analysis of each sentence within a particular segment. Each of these particular anaphoric theories would require a redefinition of the basic operations used by the semantic interpretation to manipulate discourse spaces and vertex sets. This will be discussed further in section 8.1.2 of chapter 8.

One final aspect of anaphoric constraint has yet to be discussed. That is, the implicit constraint imposed by the range or lack of different anaphoric antecedents made available by the framework. The previous discussion of anaphoric reference constraints only comes to bear once an antecedent has been made available by the framework. However, unlike the majority of anaphoric semantic theories, GTS makes available antecedents derived from sub-sentential information. An example, which shows that sub-sentential anaphoric information might be useful, is repeated below.

(258) Some farmers who own a donkey beat it. This would not happen if they were inspected by vets.

The discourse in (258) has a reading where the anaphor *they* refers to *donkeys owned by farmers* **not** just the *donkeys owned by the some farmers who beat the donkeys they own*. To this extent, GTS provides a wider range of possible antecedents than most if not all competing theories of noun phrase anaphora.

6.2.2 Principal Semantic Parameters

The GTS framework is a very general framework for the analysis of discourse anaphora. It provides a range of constraint possibilities in the representational and denotational domains for the derivation of particular theories of discourse anaphora. This section will list the available semantic "parameters" which the framework makes available. The word "parameter" is used here in a broad sense to cover the unification features which play a purely semantic role in the representational domain and anaphoric semantic functions from the semantic interpretation. The list of principal parameters is shown below.

1. Representational parameters. These parameters are all features used by the semantic interpretation which will need to be constrained through the construction of an appropriate grammar in order that only acceptable instances of the available semantic predicates are derived from the analysis of a discourse.

- uniq: This feature is utilized in both the semantic interpretation of generalized quantifiers and verbal predicates. See section 5.3.2 of chapter 5 for its use within the semantic framework.
- anaphor: This feature occurs in pronominal semantic predicates, taking two values bound and referential. An appropriate grammar may wish to constrain under which contexts a pronoun may be referential or bound. See section 6.1.5 of this chapter.
- reading: This feature determines the basic type of verbal reading given to a verbal predicate. Some verbal predicates may be constrained to have only a subset of the available readings. See section 5.3.3 of chapter 5 for its use within the framework.
- negreading: This feature determines the type of negative verbal reading. See section 5.3.3 of chapter 5.
- scope: This feature determines the scope given to generalized quantifiers within the analysis of verbal readings. See section 5.3.3 of chapter 5.
- aarel: This feature determines the type of anaphoric reading given to bound pronouns during the analysis of a verbal relation. See section 6.1.6 of this chapter.

2. Semantic interpretation parameters: the main semantic interpretation parameters are the anaphoric constraint functions which determine which anaphors and antecedents may be anaphorically connected within a discourse.

   - CONS: anaphoric constraint function: This function determines for a particular anaphor in a discourse, the set of antecedents which are acceptable, possibly none.
   - RES: anaphoric resolution function: This function determines from the set of acceptable antecedents output by the anaphoric constraint function, CONS, the set of antecedents actually referred to by the particular anaphor in question.

I have not included in the list of representational parameters, features such as pol and number which have both a syntactic and semantic role, even though these could be considered parameters in the broader sense.

By providing an appropriate grammar which determines for any sentence the allowable representational parameters and providing appropriate anaphoric constraint functions, a particular empirically sensitive theory of discourse anaphora can be derived.

In the next two sections, examples are given of constraints in the representational and denotational domains which illustrate particular partial parameterizations of the framework.

6.2.3 Representational Constraints - Kanazawa's Donkeys

Makoto Kanazawa (1994) provides evidence that the monotonicity of the subject noun phrase in donkey sentences helps determine whether a strong or weak anaphor-antecedent relation is imposed. He utilizes the concepts of left and right monotonicity, which are defined below, with respect to a model $M = \langle D, F \rangle$, where *Det* is a determiner, and $Det_D A$ is a generalized quantifier over domain *D*.

Definition 24: Left and Right Monotonicity.

A determiner, *Det*, is *right monotone decreasing* ($MON\downarrow$) if:

Monotonicity	Weak/Strong Anaphor-Antecedent Relation	Determiners
$\uparrow MON \uparrow$	Weak	a,some,several,at least n,many
$\uparrow MON \downarrow$	Strong	not every, not all
$\downarrow MON \uparrow$	Strong	every,all
$\downarrow MON \downarrow$	Weak	no,few,at most n

Table 6.1: The variations predicted by Kanazawa for strong and weak anaphor-antecedent relations.

- For all $A, B, B' \subseteq D, Det_D A, B$ and $B' \subseteq B$ imply $Det_D A, B'$.

A determiner, *Det*, is *right monotone increasing* ($MON \uparrow$) if:

- For all $A, B, B' \subseteq D, Det_D A, B$ and $B \subseteq B'$ imply $Det_D A, B'$.

A determiner, *Det*, is *left monotone decreasing* ($\downarrow MON$) if:

- For all $B, A, A' \subseteq D, Det_D A, B$ and $A' \subseteq A$ imply $Det_D A, B'$.

A determiner, *Det*, is *left monotone increasing* ($\uparrow MON$) if:

- For all $B, A, A' \subseteq D, Det_D A, B$ and $A \subseteq A'$ imply $Det_D A, B'$.

The previous definitions of monotone increasing and monotone decreasing that I have been using correspond to right monotonicity increasing and decreasing. Kanazawa's empirical theory is summarised in table 6.1. These predications can be incorporated into the GTS framework by utilizing the representational constraint possibilities available through the feature based semantic representation and its implementation within a unification grammar formalism, such as PATR. Firstly, the control feature for determiners can be extended to incorporate a feature aarel with values strong and weak. This feature will specify the preference the determiner has for either a strong or weak anaphor-antecedent relation. Following the preferences predicated by Kanazawa, as shown in table 6.1, particular semantic representations can be given for determiners. I will assume the grammar given in appendix B as given and extend this grammar. Some example PATR definitions for some unary determiners from table 6.1 are given below.

- ```
  Word every: <cat> = det
           <head sem type> = det
           <head sem control pred> = every
           <head sem control uniq> = no
           <head sem control pol> = positive
           <head sem control reading> = distributive
           <head sem control aarel> = strong
           <head sem arg1> = <subcat first head sem>
           <head syn number> = singular
           <subcat first cat> = nbar
           <subcat rest> = end.
  ```

- ```
  Word some: <cat> = det
           <head sem type> = det
  ```

```
          <head sem control pred> = some
          <head sem control uniq> = no
          <head sem control pol> = positive
          <head sem control reading> = distributive
          <head sem control aarel> = weak
          <head sem arg1> = <subcat first head sem>
          <head syn number> = singular
          <subcat first cat> = nbar
          <subcat rest> = end.
```

- `Word few: <cat> = det`

```
          <head sem type> = det
          <head sem control pred> = few
          <head sem control uniq> = no
          <head sem control pol> = positive
          <head sem control reading> = distributive
          <head sem control aarel> = weak
          <head sem arg1> = <subcat first head sem>
          <head syn number> = plural
          <subcat first cat> = nbar
          <subcat rest> = end.
```

The complex determiners *not every* and *not all* are slightly more difficult to handle within PATR in a economic manner. We require a grammar rule for these determiners, whose embryonic form is shown below.

-
```
 Det_1 -> Neg Det_2:
<Det_1 head> = <Det_2 head>
<Det_1 subcat> = <Det_2 subcat>
<Det_1 head syn> = <Det_2 head syn>
<Det_2 head sem control aarel> = strong
<Det_1 head sem control pol> = negative.
```

This PATR rule ensures that the syntactic and subcategorization information of the determiner `Det_2` is passed to `Det_1`. Furthermore, we ensure that the determiner `Det_2` is one which requires a strong anaphor-antecedent relation, thereby eliminating (as can be observed in table 6.1) possible complex negative determiners such as *not some*, *not few* or *not no*. Unfortunately, one determiner which seems to be incorrectly excluded in this is *not many*. However, the PATR rule (disregarding the mentioned inconsistency) is still incomplete, as the semantic information for `Det_1` has not been specified in full. The problem is that in the case where `Det_2` is *every* the feature `<Det_2 head sem control pol>` is `positive` and therefore we can't just transfer the semantic information across. The simple mechanisms of PATR only allow feature unification whereas what we really need to set the value of `<Det_1 head sem control pol>` to be the $\overline{xor}$(−, `<Det_2 head sem control pol>`) where $\overline{xor}$ derives the logical combination of the polarities of its arguments, as shown below.

- $\overline{xor}(+,+) = +$
  $\overline{xor}(+,-) = -$
  $\overline{xor}(-,+) = -$
  $\overline{xor}(-,-) = +$

However, such an operation is not easily derivable in PATR and thus there is no simple mechanism for handling multiple complex determiners such as *not not every*[4]. For singular occurrences we will need to transfer the rest of the semantic information across piecemeal, resulting in the final PATR rule below.

```
•    Det_1 -> Neg Det_2:
    <Det_1 head> = <Det_2 head>
    <Det_1 subcat> = <Det_2 subcat>
    <Det_1 head syn> = <Det_2 head syn>
    <Det_2 head sem control aarel> = strong
    <Det_2 head sem control pol> = positive
    <Det_1 head sem control pol> = negative
    <Det_1 head sem control reading> =
            <Det_2 head sem control reading>
    <Det_1 head sem control uniq> = <Det_2 head sem control uniq>
    <Det_1 head sem control aarel> = <Det_2 head sem control aarel>
    <Det_1 head sem pred> = <Det_2 head sem pred>
    <Det_1 head sem arg1> = <Det_2 head sem arg1>
```

The rule above ensures also the polarity of the determiner `Det_2` is `positive`, thus eliminating the possibility of multiple complex determiners such as *not not every*. We now require that verbal predicates receive their value for the feature aarel from their subject noun phrase arguments. An appropriate piece of PATR for this is shown below.

```
•   S -> NP VP:
      <S head> = <VP head>
      <S head syn form> = finite
      <VP subcat first> = <NP>
      <VP subcat rest> = end
      <S head sem control subject> = <NP head sem control>
      <S head sem control predicate aarel> =
                  <NP head sem control aarel>
      <NP head syn rel> = false.
```

These extensions to the basic grammar encode Kanazawa's predications.

6.2.4 Denotational Constraints - Semantic Number

Semantic number constraints have been advocated previously in the area of anaphoric constraint by Elworthy (1993).

The interpretation function in GTS is applied with respect to an anaphoric constraint function $\mathcal{CONS}$ which takes a triple $\langle \alpha, \mathcal{D}, G \rangle$ where α is the semantic representation for an anaphor, $\mathcal{D}$ is a discourse space and $G \in \mathcal{D}$ is a denotation graph and returns a set of *anaphor antecedent denotation pairs* $\langle \mathcal{C}, \mathcal{R} \rangle$ where $\mathcal{C}$ is a denotation set (for an anaphor) and is $\mathcal{R}$ a set of vertex graph pairs (for the antecedents). I will illustrate in this section how a $\mathcal{CONS}$ function implementing semantic constraints could be derived.

[4]It should be stated that this "problem" with PATR may more be due to the manner in which I treat multiple negations as the compositional combination of a binary `pol` feature than the limitations of PATR itself

I will look at referential pronouns first. An appropriate constraint function is provided below.

$$\mathcal{CONS}(\left[\begin{array}{ll}\textsf{control} & \left[\begin{array}{ll}\textsf{pred} & \textsf{PRO}\\ \textsf{anaphor} & \textsf{referential}\\ \textsf{number} & \textsf{NUM}\end{array}\right]\end{array}\right]_{pro}, \mathcal{D}, G) = \text{the set of } \langle \mathcal{C}, \mathcal{R} \rangle \text{ such that:}$$

- If $\textsf{NUM} = \textsf{singular}$ then $\mathcal{R} = \{\langle\langle i, \{\{x\}\}\rangle, G\rangle\}$ for some identifier i, individual x and graph G: $\mathcal{C} = \{\{x\}\}$.
- If $\textsf{NUM} = \textsf{plural}$ then $\mathcal{R} \neq \{\langle\langle i, \{\{x\}\}\rangle, G\rangle\}$ for some identifier i, individual x and graph G: $\mathcal{C} = \{X\}$ and $|X| > 1$, for some set of individuals X derived from $\mathcal{R}$.

These constraints would correctly disallow the following simple discourses where the anaphors in (259) and (260) are treated in a referential manner.

(259) Some men are walking. ?He whistles.

(260) A man is walking. ?They whistle.

A possible counterexample to these checks on referential pronouns is given below.

(261) The person who took my mug should return it immediately. They will not be punished. No penalties will be imposed upon them.

This use of (syntactically) plural pronouns to refer to singular antecedents is probably becoming more acceptable due to the desire to utilize pronouns in a gender neutral manner.

For bound pronouns simple semantic number constraints are harder to conceive. Singular bound pronouns can refer to semantically plural antecedents, as illustrated by subordination examples such as given below.

(262) Every rice-grower in Korea owns a wooden cart. He uses it when he harvests the crop. (Sells, 1986, p. 436)

Meanwhile, it is hard to know whether to place semantic number constraints on plural bound pronouns. For instance, given the following discourse in a situation where there is only a single farmer who owns a single donkey: should we disallow the plural anaphoric references of the second sentence?

(263) Every farmer owns a donkey. They beat them.

Determining whether such a constraint is required may well hinge on whether the speaker of the discourse knows that there is only a single farmer who owns a single donkey. However, even if the speaker knows this information the reference may be applicable due to similar considerations to the gender-neutral use of plural pronouns as illustrated above. Two further contrasting examples of this form are shown below.

(264) Every striker thinks he is as good as Gary Lineker.

(265) Every striker thinks they are as good as Gary Lineker.

For these reasons I will not attempt to provide a formal equivalent semantic number constraint for bound pronouns along the lines outlined for referential pronouns above.

6.3 Worked Example

I shall provide a step by step detailed worked example within this section. The two sentence discourse below will be analysed.

(266) Every farmer owns two donkeys. They beat them.

I will not provide a completely "parameterized" theory within the GTS framework in which to analyse this discourse. This will allow various decision points reached during the analysis to be illustrated.

The grammar against which this discourse will be analysed is the grammar given in appendix B. This grammar, in the derivation of semantic representations for various constituents, provides some of the required parameterization to derive an anaphoric theory from the GTS framework. The discourse will be interpreted against the model given below.

(267) $M_1 = \langle D_1, F_1 \rangle$, where
$D_1 = \{a,b,c,d,e,f,g,h\}$,
$F_1(\textsf{farmer}) = \{a,b\}$
$F_1(\textsf{donkey}) = \{d,e,f,g\}$
$F_1(\textsf{own}) = \{\langle\{a\},\{d\}\rangle,\langle\{a\},\{e\}\rangle,\langle\{b\},\{f\}\rangle,\langle\{b\},\{g\}\rangle\}$
$F_1(\textsf{beat}) = \{\langle\{a\},\{d\}\rangle,\langle\{a\},\{e\}\rangle,\langle\{b\},\{f\}\rangle,\langle\{b\},\{g\}\rangle\}$

I will assume a set of identifiers $I_1 = \{1,2,3...\}$. The anaphoric constraint function $\mathcal{CONS}$ as well as the anaphoric resolution function $\mathcal{RES}$ will not be specified. I will assume through the analysis that an appropriately realised set of $\mathcal{CONS}$ and $\mathcal{RES}$ functions exist which enforce the pronoun *they* in the second sentence to refer to the *farmers who own the donkeys* and the pronoun *them* in the second sentence to refer to the *donkeys the farmers own.* The semantic representation for the first sentence in (266) is shown below.

(268) $\alpha_1 =$

$$\begin{bmatrix} \textsf{sem} & \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{subject} & \begin{bmatrix} \textsf{reading} & \textsf{distributive} \\ \textsf{pol} & + \\ \textsf{uniq} & - \end{bmatrix} \\ \textsf{object} & \begin{bmatrix} \textsf{reading} & \textsf{distributive} \\ \textsf{pol} & + \\ \textsf{uniq} & - \end{bmatrix} \\ \textsf{predicate} & \begin{bmatrix} \textsf{pred} & \textsf{own} \\ \textsf{pol} & + \\ \textsf{scope} & \textsf{subjectwide} \\ \textsf{aarel} & \textsf{weak} \end{bmatrix} \end{bmatrix} \\ \textsf{arg1} & \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{every} \\ \textsf{uniq} & - \\ \textsf{pol} & + \end{bmatrix} \\ \textsf{arg1} & \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{farmer} \end{bmatrix} \end{bmatrix}_{n} \end{bmatrix}_{det} \\ \textsf{arg2} & \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{two} \\ \textsf{uniq} & - \\ \textsf{pol} & + \end{bmatrix} \\ \textsf{arg1} & \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{donkey} \end{bmatrix} \end{bmatrix}_{n} \end{bmatrix}_{det} \end{bmatrix}_{tv} \end{bmatrix}$$

The inductive procedure for the interpretation of a discourse was given in (253) and (254), repeated below.

(269) $[\![\alpha_1]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle((\langle\{\},\{\},\{\}\rangle,\{\}),(i,G_1,\mathcal{D}_1)\rangle$

(270) $[\![\alpha_j]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle((\langle\{\},\{\},\{\}\rangle,\mathcal{D}_{j-1}),(i,G_j,\mathcal{D}_j)\rangle$

From this we can see that we begin the analysis with an empty discourse space and graph. The interpretation of α_1 itself is compositional in nature and follows the principle discussed in section 4.4 of chapter 4. The interpretation threads the input graph and discourse space through the analysis of the constituents building up a graph for the entire sentence. Each analysed constituent derives a graph which is placed in the discourse space. Via this compositional interpretation the first constituent to be fully analysed will be that of the nominal predicate farmer. The interpretation of lexical nouns given in section 6.1.3 is repeated below.

$$\text{If} \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{A} \\ \textsf{number} & \textsf{B} \end{bmatrix} \end{bmatrix}_n \sqsubseteq \alpha \text{ and } \begin{pmatrix} \textsf{PRED} = \alpha/\langle\textsf{control pred}\rangle \\ \textsf{NUM} = \alpha/\langle\textsf{control number}\rangle \end{pmatrix} \text{ then}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(G,\mathcal{D}),(i,G[v],\mathcal{D}\cup\{G[v]\})\rangle$, where,

- **[1]** $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$.
- **[2]** $v = \langle i, \mathcal{C}\rangle$ where $\mathcal{C} = \begin{cases} \{X \subseteq F(\textsf{PRED}) \,|\, |X| = 1\} & \text{If } \textsf{NUM} = \textsf{singular} \\ \{X \subseteq F(\textsf{PRED}) \,|\, |X| \geq 1\} & \text{If } \textsf{NUM} = \textsf{plural} \end{cases}$

I have numbered each interpretation rule so that they can be referred to in the following text. The values of various structures when applying this interpretation rule are shown below.

- PRED = farmer
- NUM = singular
- $G = \langle\{\},\{\},\{\}\rangle$ (an empty graph)
- $\mathcal{D} = \{\}$

From **[1]** we need a new unique identifier: let this be 1. In **[2]** we create a new vertex describing the individuals that satisfy the nominal predicate. The value of the denotation set $\mathcal{C}$ derived is given below.

- $\mathcal{C} = \{\{a\},\{b\}\}$ (The singleton sets of farmers)

The output graph and discourse space are illustrated in figure 6.5. The next feature structure to be interpreted will be for the determiner predicate every. The interpretation of determiner predicates (generalized quantifiers) was given in section 6.1.4 and is repeated below.

$$\text{If} \begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{pred} & \textsf{A} \\ \textsf{number} & \textsf{B} \\ \textsf{uniq} & \textsf{C} \\ \textsf{pol} & \textsf{D} \end{bmatrix} \\ \textsf{arg1} & \textsf{G} \end{bmatrix}_{det} \sqsubseteq \alpha \text{ and } \begin{pmatrix} \textsf{PRED} = \alpha/\langle\textsf{control pred}\rangle \\ \textsf{NUM} = \alpha/\langle\textsf{control number}\rangle \\ \textsf{ARG} = \alpha/\langle\textsf{arg1}\rangle \end{pmatrix} \text{ then}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(G,\mathcal{D}),(i,G'',\mathcal{D}'\cup\{G''\})\rangle$ where

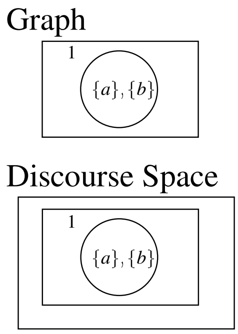


Figure 6.5: Output graph and discourse space after the analysis of the nominal predicate farmer in (268).

- **[1]** $[\![\mathsf{ARG}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$
- **[2]** $\langle i,\mathcal{C}\rangle \in G'$, $S = \bigcup_{X\in\mathcal{C}} X$
  The vertex holding the information of the argument to the determiner is $\langle i,\mathcal{C}\rangle$. We take the union of the sets in $\mathcal{C}$.
- **[3]** $\mathcal{C}' = \begin{cases} \{X \subseteq S | X = S\} & \text{If PRED = every} \\ \{X \subseteq S || X| \geq \frac{1}{2}|S|\} & \text{If PRED = most} \\ \{X \subseteq S || X| = 1\} & \text{If PRED = a} \\ \{X \subseteq S || X| = 1\} & \text{If PRED = some, NUM = singular} \\ \{X \subseteq S || X| > 1\} & \text{If PRED = some, NUM = plural} \\ \{X \subseteq S | X = S\} & \text{If PRED = no} \\ \{X \subseteq S || X| \geq \frac{1}{2}|S|\} & \text{If PRED = few} \\ \{X \subseteq S || X| = 2\} & \text{If PRED = two} \end{cases}$
  From the set S we can determine the denotation set $\mathcal{C}'$ containing the witness sets for the particular quantifier in question.
- **[4]** $G'' = G'[\langle i,\mathcal{C}'\rangle / \langle i,\mathcal{C}\rangle]$
  The graph G'' is the graph G' with the vertex $\langle i,\mathcal{C}\rangle$ replaced by the vertex $\langle i,\mathcal{C}'\rangle$.

The values of various structures when applying this interpretation rule are shown below.

- PRED = every
- NUM = singular
- $G = \langle \{\},\{\},\{\}\rangle$ (an empty graph)
- $\mathcal{D} = \{\}$
- *G' as shown in figure 6.5*
- *$\mathcal{D}'$ as shown in figure 6.5*

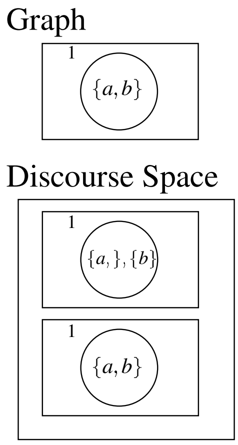


Figure 6.6: Output graph and discourse space after the analysis of the determiner predicate every in (268).

- $i = 1$
- $\mathcal{C} = \{\{a\},\{b\}\}$

In **[1]** the analysis of the nominal predicate is determined, the values of the input and output to the interpretation function are given above. In **[2]** the vertex identified by i is given and the sets from this vertex are unioned together, to give a set S, whose value is given below.

- $S = \{a,b\}$

We then use S to derive the appropriate witness sets for the quantifier, as shown in **[3]**, where $\mathcal{C}'$ is as shown below.

- $\mathcal{C}' = \{\{a,b\}\}$

The newly derived graph G'' and associated discourse space are shown in figure 6.6.

The next feature structure interpreted will be that for the lexical noun predicate donkey. Again we need to apply the interpretation rule for lexical noun predicates repeated below.

If $\left[\ \text{control} \left[\begin{array}{ll} \text{pred} & \text{A} \\ \text{number} & \text{B} \end{array}\right]\ \right]_n \sqsubseteq \alpha$ and $\left(\begin{array}{l} \text{PRED} = \alpha/\langle\text{control pred}\rangle \\ \text{NUM} = \alpha/\langle\text{control number}\rangle \end{array}\right)$ then $[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(G,\mathcal{D}),(i,G[v],\mathcal{D}\cup\{G[v]\})\rangle$, where,

- **[1]** $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$.
- **[2]** $v = \langle i,\mathcal{C}\rangle$ where $\mathcal{C} = \left\{\begin{array}{ll} \{X \subseteq F(\text{PRED}) \mid |X| = 1\} & \text{If NUM} = \text{singular} \\ \{X \subseteq F(\text{PRED}) \mid |X| \geq 1\} & \text{If NUM} = \text{plural} \end{array}\right.$

The values of various structures when applying this interpretation rule are shown below.

- PRED = donkey
- NUM = plural

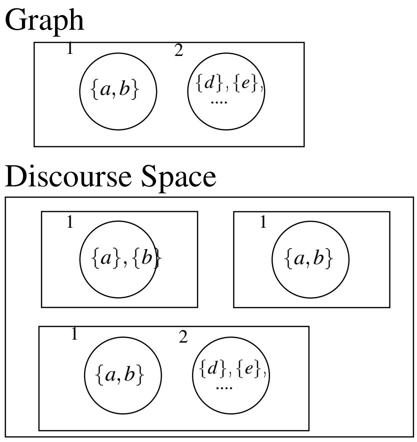


Figure 6.7: Output graph and discourse space after the analysis of the nominal predicate donkey in (268).

- G and $\mathcal{D}$ *as shown in figure 6.6*

A new identifier is required in **[1]**, let us assume this is 2. In **[2]**, a new vertex is created. The value of $\mathcal{C}$ is given below, (where $\wp$ is the power set operator).

- $\mathcal{C} = \wp\{d,e,f,g\}$

The derived graph and discourse space are illustrated in figure 6.7.

The next feature structure interpreted will be that of the determiner predicate two. The appropriate interpretation rule for determiner predicates is repeated below.

If $\begin{bmatrix} \text{control} & \begin{bmatrix} \text{pred} & \text{A} \\ \text{number} & \text{B} \\ \text{uniq} & \text{C} \\ \text{pol} & \text{D} \end{bmatrix} \\ \text{arg1} & \text{G} \end{bmatrix}_{det} \sqsubseteq \alpha$ and $\begin{pmatrix} \text{PRED} = \alpha/\langle\text{control pred}\rangle \\ \text{NUM} = \alpha/\langle\text{control number}\rangle \\ \text{ARG} = \alpha/\langle\text{arg1}\rangle \end{pmatrix}$ then

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(G,\mathcal{D}),(i,G'',\mathcal{D}'\cup\{G''\})\rangle$ where

- **[1]** $[\![\text{ARG}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$
- **[2]** $\langle i,\mathcal{C}\rangle \in G'$, $S = \bigcup_{X\in\mathcal{C}} X$
  The vertex holding the information of the argument to the determiner is $\langle i,\mathcal{C}\rangle$. We take the union of the sets in $\mathcal{C}$.

- **[3]** $C' = \begin{cases} \{X \subseteq S | X = S\} & \text{If PRED = every} \\ \{X \subseteq S | |X| \geq \frac{1}{2}|S|\} & \text{If PRED = most} \\ \{X \subseteq S | |X| = 1\} & \text{If PRED = a} \\ \{X \subseteq S | |X| = 1\} & \text{If PRED = some, NUM = singular} \\ \{X \subseteq S | |X| > 1\} & \text{If PRED = some, NUM = plural} \\ \{X \subseteq S | X = S\} & \text{If PRED = no} \\ \{X \subseteq S | |X| \geq \frac{1}{2}|S|\} & \text{If PRED = few} \\ \{X \subseteq S | |X| = 2\} & \text{If PRED = two} \end{cases}$

  From the set S we can determine the denotation set C' containing the witness sets for the particular quantifier in question.

- **[4]** $G'' = G'[\langle i, C' \rangle / \langle i, C \rangle]$

  The graph G'' is the graph G' with the vertex $\langle i, C \rangle$ replaced by the vertex $\langle i, C' \rangle$.

The values of various structures when applying this interpretation rule are shown below.

- PRED = two
- NUM = plural
- G and $\mathcal{D}$ *as shown in figure 6.6*
- G' *as shown in figure 6.7*
- $\mathcal{D}'$ *as shown in figure 6.7*
- $i = 2$
- $C = \wp\{d, e, f, g\}$

In **[1]** the analysis of the nominal predicate is determined, the values of the input and output to the interpretation function are given above. In **[2]** the vertex identified by i is given and the sets from this vertex are unioned together, to give a set S, whose value is given below.

- $S = \{d, e, f, g\}$

We then use S to derive the appropriate witness sets for the quantifier, as shown in **[3]**, where C' is as shown below.

- $C' = \{\{d, e\}, \{d, f\}, \{e, f\}, \{d, g\}, \{e, g\}, \{f, g\}, \{d, e, f\}, \{d, e, g\}, \{d, f, g\}, \{e, f, g\}, \{d, e, f, g\}\}$

The newly derived graph G'' and associated discourse space are shown in figure 6.8.

The next feature structure interpreted would be that of the transitive verbal predicate own. The interpretation of transitive verbal predicates given previously in section 6.1.6 is repeated below.

$$\text{If} \begin{bmatrix} \text{control} & \begin{bmatrix} \text{subject} & \text{S} \\ \text{object} & \text{O} \\ \text{predicate} & \text{P} \end{bmatrix} \\ \text{arg1} & \text{X} \\ \text{arg2} & \text{Y} \end{bmatrix}_{tv} \sqsubseteq \alpha, \text{ and } \begin{pmatrix} \text{V} = \alpha/\langle\text{control predicate pred}\rangle \\ \text{ARG1} = \alpha/\langle\text{arg1}\rangle \\ \text{ARG2} = \alpha/\langle\text{arg2}\rangle \\ \text{CTRL} = \alpha/\langle\text{control}\rangle \\ \text{AAREL} = \alpha/\langle\text{control aarel}\rangle \end{pmatrix} \text{ then,}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1, \mathcal{D}_1), (i_1, G_4, \mathcal{D}_3 \cup \{G_4\}) \rangle$ where

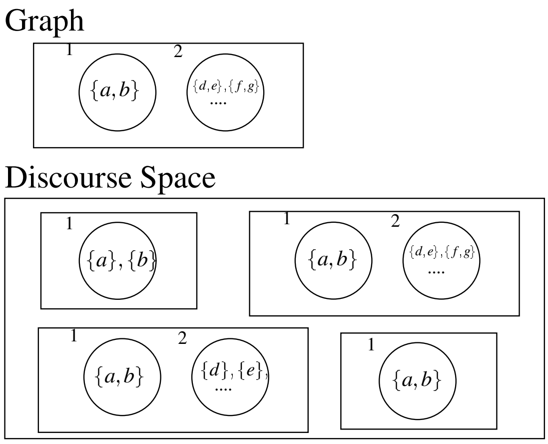


Figure 6.8: Output graph and discourse space after the analysis of the determiner predicate two in (268).

- **[1]** $[\![\text{ARG1}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1, \mathcal{D}_1), (i_1, G_2, \mathcal{D}_2) \rangle$
- **[2]** $[\![\text{ARG2}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_2, \mathcal{D}_2), (i_2, G_3, \mathcal{D}_3) \rangle$
- **[3]** $v = \langle i_1, \mathcal{C} \rangle$ where $\langle i_1, \mathcal{C} \rangle \in G_3$ and $v' = \langle i_2, \mathcal{C}' \rangle$ where $\langle i_2, \mathcal{C}' \rangle \in G_3$
  The vertices for each argument are determined via the identifiers i_1 and i_2.
- **[4]** $R = \{\langle X, Y \rangle | \exists S_1 \in \mathcal{C}, \exists S_2 \in \mathcal{C}' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The relation R allows any pair of subsets from either argument.
- **[5]** $R_a = \{\langle X, Y \rangle \in R | satis(G_3[\langle v, v', R \rangle], \mathcal{L}) \wedge \{\langle v, X \rangle, \langle v', Y \rangle\} \subseteq \mathcal{L}\}$
  The relation R_a limits the relation R by allowing only anaphorically acceptable pairs from R. This is determined via the relation $satis$ (defined on page 86) over the graph G extended with an edge between the vertices v and v′ utilizing the relation R. The sets X and Y are labels for the vertices v and v′ respectively.
- **[6]** If ϕ is the interpretation rule derived from CTRL then:
  - If SCOPE = subjectwide, $\phi' = (((\phi(\mathcal{C})), (\mathcal{C}')), (F(V)))$
  - If SCOPE = objectwide, $\phi' = (((\phi(\mathcal{C}')), (\mathcal{C})), (F(V)))$

  The verbal reading rule is derived, the arguments to ϕ being given in an order determined by the feature scope.
- **[7]** There is a mapping from $\langle \mathcal{C}, \mathcal{C}', R_a \rangle$ to $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ where $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ is the component-wise union of all triples, $\langle \mathcal{C}_t, \mathcal{C}_t', R_t \rangle, \mathcal{C}_t \subseteq \mathcal{C}, \mathcal{C}_t' \subseteq \mathcal{C}', R_t \subseteq R_a$ which minimally satisfy the interpretation rule ϕ'
  The set of triples $\langle \mathcal{C}_t, \mathcal{C}_t', R_t \rangle$ which satisfy the verbal reading described by ϕ' are collected together in $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$.

- **[8]** $v_s = \langle i_1, \mathcal{C}_s \rangle$ and $v_o = \langle i_2, \mathcal{C}_o \rangle$.
  New vertices are constructed.

- **[9]** If $R' = \{\}$ then $G_4 = cons(G_3[v_s/v, v_o/v'])$ else $G_4 = cons(G_3[[v_s/v, v_o/v'][\langle v_s, v_o, R' \rangle])$
  If the derived relation R' is empty no relational edge is constructed between the new vertices. The function cons (defined on page 86) forces the new graphs to be maximally consistent.

In **[1]** and **[2]** the arguments are interpreted. The interpretation of the two arguments have been shown above. The values of the structures derived are given below.

- $G_1 = \langle \{\}, \{\}, \{\} \rangle$
- $\mathcal{D}_1 = \{\}$
- G_2 and $\mathcal{D}_2$ *as shown in figure 6.6*
- G_3 and $\mathcal{D}_3$ *as shown in figure 6.8*

In **[3]**, the two vertices describing the arguments are extracted. The values of the vertex structures are given below.

- $i_1 = 1$
- $\mathcal{C} = \{\{a, b\}\}$
- $i_2 = 2$
- $\mathcal{C}' = \{\{d,e\}, \{d,f\}, \{e,f\}, \{d,g\}, \{e,g\}, \{f,g\}, \{d,e,f\}, \{d,e,g\}, \{d,f,g\}, \{e,f,g\}, \{d,e,f,g\}\}$

In **[4]**, a unrestricted relation R is derived. This relation pairs every subset of every set in $\mathcal{C}$ with every subset of every set in $\mathcal{C}'$. This relation R is partially given below.

- $R = \{\langle \{a\}, \{d\} \rangle, \langle \{a\}, \{e\} \rangle, \langle \{a\}, \{f\} \rangle, \langle \{b\}, \{d\} \rangle, \langle \{b\}, \{e\} \rangle, \langle \{b\}, \{f\} \rangle, \langle \{a,b\}, \{d\} \rangle, \langle \{a,b\}, \{e\} \rangle, \langle \{a,b\}, \{f\} \rangle, ...\}$

In **[5]** we limit this relation to only those pairs of sets which are anaphorically acceptable. An anaphorically acceptable pair of sets is a pair which contributes to a globally satisfiable labelling of the constraint network derived by extending the graph G_3 with a relationally edge defined by R over the two argument vertices. As the two argument vertices are the only vertices in this graph and the relation R is the only relation, no restriction will take place and in this case $R_a = R$.

In **[5]** we derive the interpretation rule ϕ from the control feature of the verbal predicate. This control feature is repeated below.

$$\bullet \left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll} \text{subject} & \left[\begin{array}{ll} \text{reading} & \text{distributive} \\ \text{pol} & + \\ \text{uniq} & - \end{array}\right] \\ \text{object} & \left[\begin{array}{ll} \text{reading} & \text{distributive} \\ \text{pol} & + \\ \text{uniq} & - \end{array}\right] \\ \text{predicate} & \left[\begin{array}{ll} \text{pred} & \text{own} \\ \text{pol} & + \\ \text{scope} & \text{subjectwide} \end{array}\right] \end{array}\right] \end{array}\right]$$

The subject, object and verbal rules derived are shown below.

- Subject Rule = $\lambda P, C_1.\exists S_1 \in C_1 : \forall x \in S_1 \rightarrow P(\{x\})$
- Object Rule = $\lambda P', C_2.\exists S_2 \in C_2 : \forall y \in S_2 \rightarrow P'(\{y\})$
- Verbal Rule = $\lambda S_3, S_4, V.\langle S_3, S_4\rangle \in V$

These rules are combined to form the complete interpretation rule shown in (271) and its lambda reduced equivalent in (272).

(271) $\lambda P, C_1.\exists S_1 \in C_1 : \forall x \in S_1 \rightarrow P(\{x\})(\lambda P', C_2.\exists S_2 \in C_2 : \forall y \in S_2 \rightarrow P'(\{y\})(\lambda S_3, S_4, V.\langle S_3, S_4\rangle \in V))$

(272) $\lambda C_1, C_2, V.\exists S_1 \in C_1 : \forall x \in S_1 \rightarrow \exists S_2 \in C_2 : \forall y \in S_2 \rightarrow \langle\{x\},\{y\}\rangle \in V$

In **[6]**, the rule in (272) is applied to arguments C, C' and $F(\text{own})$ to give the rule shown below.

- $\lambda \exists S_1 \in C : \forall x \in S_1 \rightarrow \exists S_2 \in C' : \forall y \in S_2 \rightarrow \langle\{x\},\{y\}\rangle \in F(\text{own})$

In **[7]** we find triples consisting of subsets of C_t, C'_t and R_t of C, C' and R respectively which satisfy the interpretation rule. In this simple example there is only one such triple which satisfies this distributive interpretation and it is shown below.

- $C_t = \{\{a,b\}\}$
- $C'_t = \{\{d,e\},\{f,g\}\}$
- $R_t = \{\langle\{a\},\{d\}\rangle,\langle\{a\},\{e\}\rangle,\langle\{b\},\{f\}\rangle,\langle\{b\},\{g\}\rangle\}$

Any other triple will either not satisfy the interpretation rule or will not be minimal as described in section 5.3.3 of chapter 5. Given that there is only one satisfying triple, $R' = R_t, C_s = C_t$ and $C_o = C'_t$.

In **[8]** we create new vertices for each of the verbal argument vertices and in **[9]** we derive the new graph G_4 which as R' is nonempty will have a relational edge between v_s and v_o. This new graph is illustrated in figure 6.9. We have now completed the interpretation of the first sentence in the discourse shown in (266) and have created a discourse space containing 5 graphs derived from the interpretation of different constituents (described by semantic predicates) in the first sentence of (266).

The second sentence in (266) will have a semantic representation shown below.

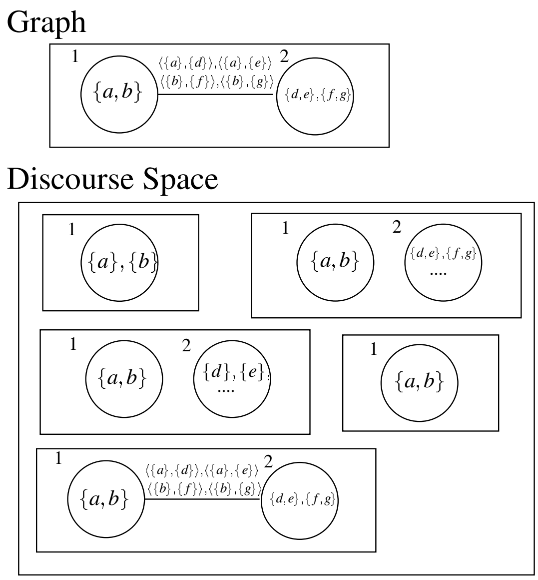


Figure 6.9: Output graph and discourse space after the analysis of the transitive verbal predicate own in (268).

(273) $\alpha_1 =$

$$\begin{array}{ll} \text{sem} & \left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll} \text{subject} & \left[\begin{array}{ll} \text{reading} & \text{distributive} \\ \text{pol} & + \\ \text{uniq} & - \end{array}\right] \\ \text{object} & \left[\begin{array}{ll} \text{reading} & \text{distributive} \\ \text{pol} & + \\ \text{uniq} & - \end{array}\right] \\ \text{predicate} & \left[\begin{array}{ll} \text{pred} & \text{beat} \\ \text{pol} & + \\ \text{scope} & \text{subjectwide} \\ \text{aarel} & \text{weak} \end{array}\right] \end{array}\right] \\ \text{arg1} & \left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll} \text{pred} & \text{they} \\ \text{anaphor} & \text{bound} \\ \text{number} & \text{plural} \end{array}\right] \end{array}\right]_{pro} \\ \text{arg2} & \left[\begin{array}{ll} \text{control} & \left[\begin{array}{ll} \text{pred} & \text{them} \\ \text{anaphor} & \text{bound} \\ \text{number} & \text{plural} \end{array}\right] \end{array}\right]_{pro} \end{array}\right]_{tv} \end{array}$$

Following our inductive procedure for analysing discourse we feed the discourse space derived from the first sentence into the interpretation of the second. The first semantic predicate to be fully analysed will be that of the pronominal predicate they. The analysis of bound pronouns is given in section 6.1.5 and is repeated below.

$$\text{If} \left[\text{control} \left[\begin{array}{ll} \text{pred} & \text{PRO} \\ \text{anaphor} & \text{bound} \\ \text{number} & \text{NUM} \end{array} \right] \right]_{pro} \sqsubseteq \alpha \text{ then}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G, \mathcal{D}), (i, G', \mathcal{D} \cup \{G'\}) \rangle$ where

- **[1]** $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$
- **[2]** $\mathcal{RES}(\mathcal{DIS}, \mathcal{CONS}(\alpha, \mathcal{D}, G)) = \langle \mathcal{C}, \mathcal{R} \rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function $\mathcal{RES}$ to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{CONS}$.
- **[3]** $v = \langle i, \mathcal{C} \rangle$.
  The vertex for the anaphor is created.
- **[4]** $A = \{\langle v, v_i \rangle | \langle v_i, G \rangle \in \mathcal{R}\}$ and $G_{t1} = \langle \{\}, \{\}, A \rangle$
  A is the set of anaphoric edges linking anaphor to antecedent, and a graph G_{t1} is created to hold these anaphoric edges.
- **[5]** $G_{t2} = \bigcup_{G \in \mathcal{R}_G} G$
  A graph G_{t2} is created from the union of the antecedent graphs.
- **[6]** $G' = G[v] \cup G_{t1} \cup G_{t2}$
  The graph for the anaphor is the union of the extension of the graph G with the anaphor vertex along with the graphs G_{t1} and G_{t2}.

The values of various structures when applying this interpretation rule are shown below.

- $G = \langle \{\}, \{\}, \{\} \rangle$
- *$\mathcal{D}$ as shown in figure 6.9*

In **[1]** we obtain a unique identifier, in this case let us assume 3. In **[2]** we derive a denotation set and a set of vertex graph pairs from application of the anaphoric resolution function to the anaphoric constraint function. In order for the desired reading we require the pronoun *they* to refer to the farmers who own the donkeys from the first sentence in (266). The desired vertex is vertex 1 of the graph shown in figure 6.9: the graph derived from the analysis of the first sentence. The following structure would have been derived.

- $\mathcal{C} = \{\{a, b\}\}$
- $\mathcal{R} = \{\langle v_a, G_a \rangle\}$ where G_a is the graph given in figure 6.9 and v_a is vertex 1 from this graph.

Note that the framework says very little about the relation of the denotation set $\mathcal{C}$ for the anaphor and the denotation set from vertex 1 of the antecedent graph. I have assumed here that we want them to have the same structure, i.e. one set containing the two individuals a and b. However, the

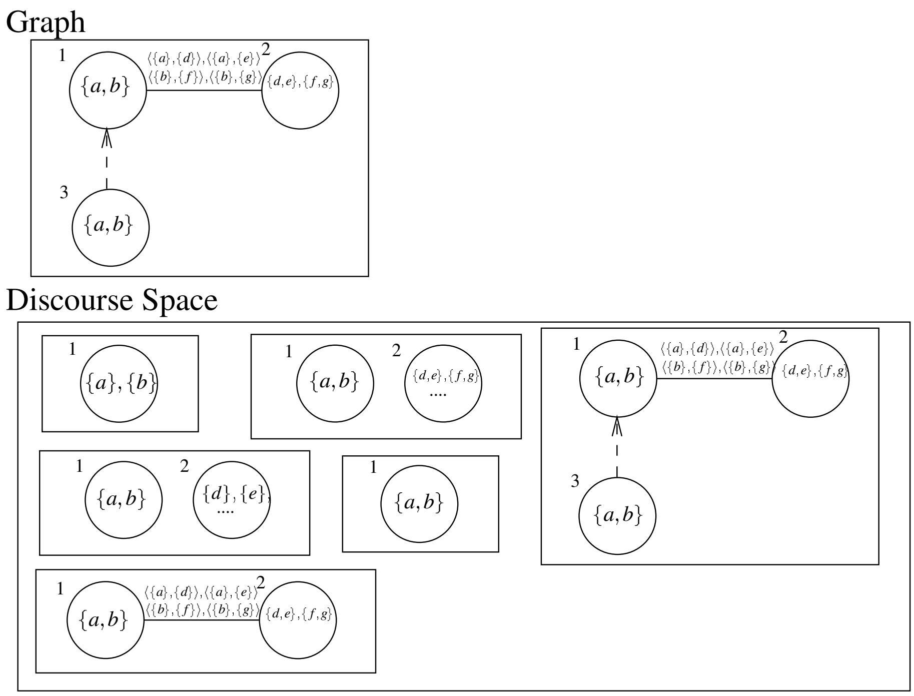


Figure 6.10: Output graph and discourse space after the analysis of the pronominal predicate they in (273).

framework does not require this. It is up to the anaphoric functions *RES* and *CONS* to specify the relationship between the denotation set for the antecedent and that for the anaphor.

In **[3]** we create a new vertex for the anaphor. In **[4]** we derive anaphoric edges linking anaphor and antecedent vertices. In this case a single anaphoric vertex between v and v_a. In **[5]** we collect the graphs of the antecedents together. In this case only the single graph shown in figure 6.9. Finally in **[6]**, we create a new graph from the union of the graphs just created with that of the input graph extended with the new vertex. The newly derived graph and resulting discourse space is shown in figure 6.10.

The next feature structure to be interpreted is that of the pronominal predicate them. The interpretation rule is repeated again.

$$\text{If} \left[\text{control} \left[\begin{array}{ll} \text{pred} & \text{PRO} \\ \text{anaphor} & \text{bound} \\ \text{number} & \text{NUM} \end{array} \right] \right]_{pro} \sqsubseteq \alpha \text{ then}$$

$$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G',\mathcal{D} \cup \{G'\}) \rangle \text{ where}$$

- **[1]** $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$

- **[2]** $\mathcal{RES}(\mathcal{DIS}, \mathcal{CONS}(\alpha, \mathcal{D}, G)) = \langle \mathcal{C}, \mathcal{R} \rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function $\mathcal{RES}$ to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{CONS}$.

- **[3]** $v = \langle i, \mathcal{C} \rangle$.
  The vertex for the anaphor is created.

- **[4]** $A = \{\langle v, v_i \rangle | \langle v_i, G \rangle \in \mathcal{R}\}$ and $G_{t1} = \langle \{\}, \{\}, A \rangle$
  A is the set of anaphoric edges linking anaphor to antecedent, and a graph G_{t1} is created to hold these anaphoric edges.

- **[5]** $G_{t2} = \bigcup_{G \in \mathcal{R}_G} G$
  A graph G_{t2} is created from the union of the antecedent graphs.

- **[6]** $G' = G[v] \cup G_{t1} \cup G_{t2}$
  The graph for the anaphor is the union of the extension of the graph G with the anaphor vertex along with the graphs G_{t1} and G_{t2}.

The values of various structures when applying this interpretation rule are shown below.

- G and $\mathcal{D}$ *as shown in figure 6.10*

In **[1]** we obtain a unique identifier, in this case let us assume 4. In **[2]** we derive a denotation set and a set of vertex graph pairs from application of the anaphoric resolution function to the anaphoric constraint function. In order for the desired reading we require the pronoun *them* to refer to the donkeys owned by the farmers from the first sentence in (266). The desired vertex is vertex 2 of the graph shown in figure 6.9: the graph derived from the analysis of the first sentence. The following structure would be derived.

- $\mathcal{C} = \{\{d, e\}, \{f, g\}\}$

- $\mathcal{R} = \{\langle v_a, G_a \rangle\}$ where G_a is the graph given in figure 6.9 and v_a is vertex 2 from this graph.

In **[3]** we create a new vertex for the anaphor. In **[4]** we derived anaphoric edges linking anaphor and antecedent vertices. In this case a single anaphoric vertex between v and v_a. In **[5]** we collect the graphs of the antecedents together. In this case only the single graph shown in figure 6.9. Finally in **[6]**, we create a new graph from the union of the graphs just created with that of the input graph extended with the new vertex. The newly derived graph and resulting discourse space is shown in figure 6.11. Note, that as the antecedent graph has already been incorporated into G from the analysis of the pronominal predicate *them* it will **not** appear twice in G' as a graph is made up of a **set** of vertices, relations and anaphoric edges.

Having interpreted the two arguments to the verbal predicate beat we return to analyse the predicate itself. The interpretation rule for verbal predicates is repeated below.

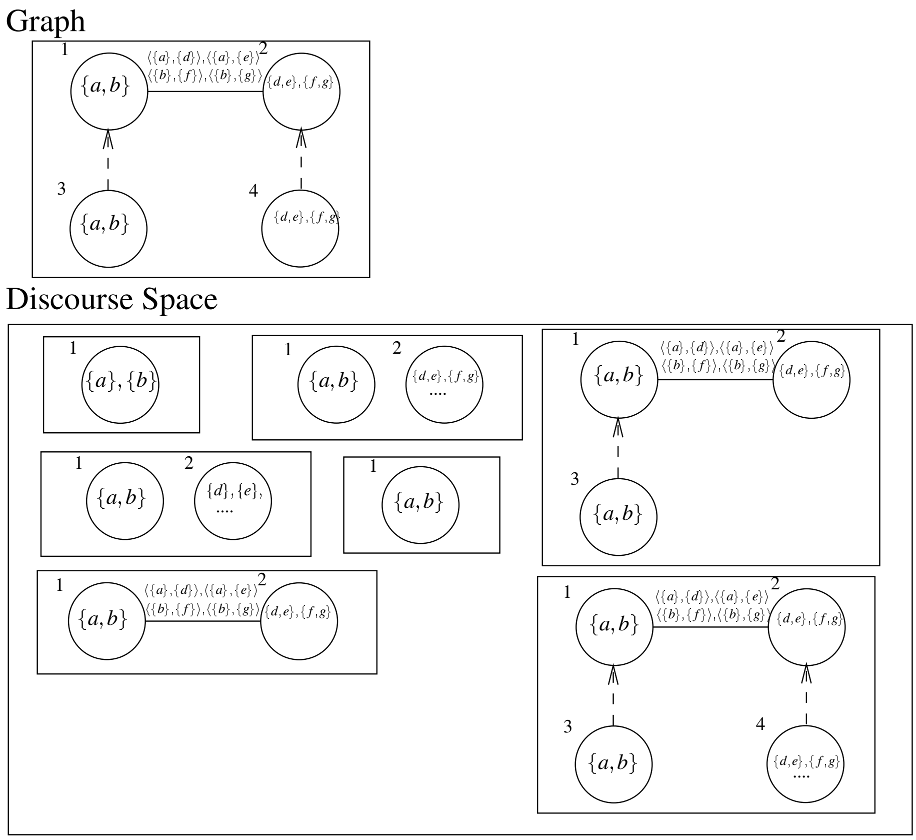


Figure 6.11: Output graph and discourse space after the analysis of the pronominal predicate them in (273).

If $\begin{bmatrix} \textsf{control} & \begin{bmatrix} \textsf{subject} & \textsf{S} \\ \textsf{object} & \textsf{O} \\ \textsf{predicate} & \textsf{P} \end{bmatrix} \\ \textsf{arg1} & \textsf{X} \\ \textsf{arg2} & \textsf{Y} \end{bmatrix}_{tv} \sqsubseteq \alpha$, and $\begin{pmatrix} \textsf{V} = \alpha/\langle \textsf{control predicate pred} \rangle \\ \textsf{ARG1} = \alpha/\langle \textsf{arg1} \rangle \\ \textsf{ARG2} = \alpha/\langle \textsf{arg2} \rangle \\ \textsf{CTRL} = \alpha/\langle \textsf{control} \rangle \\ \textsf{AAREL} = \alpha/\langle \textsf{control aarel} \rangle \end{pmatrix}$ then,

$[[\alpha]]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1, \mathcal{D}_1), (i_1, G_4, \mathcal{D}_3 \cup \{G_4\}) \rangle$ where

- **[1]** $[[\textsf{ARG1}]]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1, \mathcal{D}_1), (i_1, G_2, \mathcal{D}_2) \rangle$
- **[2]** $[[\textsf{ARG2}]]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_2, \mathcal{D}_2), (i_2, G_3, \mathcal{D}_3) \rangle$
- **[3]** $v = \langle i_1, \mathcal{C} \rangle$ where $\langle i_1, \mathcal{C} \rangle \in G_3$ and $v' = \langle i_2, \mathcal{C}' \rangle$ where $\langle i_2, \mathcal{C}' \rangle \in G_3$
  The vertices for each argument are determined via the identifiers i_1 and i_2.
- **[4]** $R = \{\langle X, Y \rangle | \exists S_1 \in \mathcal{C}, \exists S_2 \in \mathcal{C}' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The relation R allows any pair of subsets from either argument.
- **[5]** $R_a = \{\langle X, Y \rangle \in R | satis(G_3[\langle v, v', R \rangle], \mathcal{L}) \wedge \{\langle v, X \rangle, \langle v', Y \rangle\} \subseteq \mathcal{L}\}$
  The relation R_a limits the relation R by allowing only anaphorically acceptable pairs from R. This is determined via the relation satis (defined on page 86) over the graph G extended with an edge between the vertices v and v′ utilizing the relation R. The sets X and Y are labels for the vertices v and v′ respectively.
- **[6]** If ϕ is the interpretation rule derived from CTRL then:
  - If SCOPE = subjectwide, $\phi' = (((\phi(\mathcal{C})), (\mathcal{C}')), (F(V)))$
  - If SCOPE = objectwide, $\phi' = (((\phi(\mathcal{C}')), (\mathcal{C})), (F(V)))$

  The verbal reading rule is derived, the arguments to ϕ being given in an order determined by the feature scope.
- **[7]** There is a mapping from $\langle \mathcal{C}, \mathcal{C}', R_a \rangle$ to $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ where $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ is the component-wise union of all triples, $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle, \mathcal{C}_t \subseteq \mathcal{C}, \mathcal{C}'_t \subseteq \mathcal{C}', R_t \subseteq R_a$ which minimally satisfy the interpretation rule ϕ'
  The set of triples $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle$ which satisfy the verbal reading described by ϕ' are collected together in $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$.
- **[8]** $v_s = \langle i_1, \mathcal{C}_s \rangle$ and $v_o = \langle i_2, \mathcal{C}_o \rangle$.
  New vertices are constructed.
- **[9]** If $R' = \{\}$ then $G_4 = cons(G_3[v_s/v, v_o/v'])$ else
  $G_4 = cons(G_3[[v_s/v, v_o/v'][\langle v_s, v_o, R' \rangle]])$
  If the derived relation R' is empty no relational edge is constructed between the new vertices. The function cons (defined on page 86) forces the new graphs to be maximally consistent.

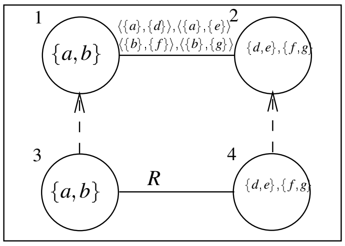


Figure 6.12: The graph used to determine anaphorically-acceptable labellings during the analysis of the verbal predicate beat

In **[1]** and **[2]** the arguments are interpreted. The interpretation of the two arguments have been shown above. The values of the structures derived are given below.

- $G_1 = \langle \{\}, \{\}, \{\} \rangle$
- $\mathcal{D}_1$ *as shown in figure 6.9*
- G_2 and $\mathcal{D}_2$ *as shown in figure 6.10*
- G_3 and $\mathcal{D}_3$ *as shown in figure 6.11*

In **[3]**, the two vertices describing the arguments are extracted. The values of the vertex structures are given below.

- $i_1 = 3$
- $\mathcal{C} = \{\{a,b\}\}$
- $i_2 = 4$
- $\mathcal{C}' = \{\{d,e\},\{f,g\}\}$

In **[4]**, a unrestricted relation R is derived. This relation pairs every subset of every set in $\mathcal{C}$ with every subset of every set in $\mathcal{C}'$. This relation R is given below.

- $R = \{\langle\{a\},\{d\}\rangle, \langle\{a\},\{e\}\rangle, \langle\{a\},\{f\}\rangle, \langle\{a\},\{g\}\rangle, \langle\{b\},\{d\}\rangle, \langle\{b\},\{e\}\rangle, \langle\{b\},\{f\}\rangle, \langle\{b\},\{g\}\rangle, \langle\{a,b\},\{d\}\rangle, \langle\{a,b\},\{e\}\rangle, \langle\{a,b\},\{f\}\rangle, \langle\{a,b\},\{g\}\rangle, \langle\{a,b\},\{d,e\}\rangle, \langle\{a,b\},\{f,g\}\rangle, \langle\{a\},\{f,g\}\rangle, \langle\{b\},\{f,g\}\rangle\}$

In **[5]** we limit this relation to only those pairs of sets which are anaphorically acceptable. An anaphorically acceptable pair of sets is a pair which contributes to a globally satisfiable labelling of the constraint network derived by extending the graph G_3 with a relationally edge defined by R over the two argument vertices. This graph is shown in figure 6.12. There is only one anaphoric circuit in this graph, described by the vertices 1,2,4,3. By treating this graph as a constraint network we can limit the relation R to contain only anaphorically acceptable pairs. The relation R_a will be as follows.

- $R_a = \{\langle\{a\},\{d\}\rangle, \langle\{a\},\{e\}\rangle, \langle\{b\},\{f\}\rangle, \langle\{b\},\{g\}\rangle\}$

In **[6]** we create the appropriate verbal rule which is identical to the rule created for the analysis of the verbal predicate own of the previous sentence, i.e. a subject and object distributive positive polarity reading with no uniqueness conditions and subjectwide scope. The derived rule is shown below.

- $\lambda C_1, C_2, V.\exists S_1 \in C_1 : \forall x \in S_1 \rightarrow \exists S_2 \in C_2 : \forall y \in S_2 \rightarrow \langle\{x\},\{y\}\rangle \in V$

In **[6]**, the rule in (6.3) is applied to arguments $\mathcal{C}$, $\mathcal{C}'$ and $F(\mathsf{beat})$ to give the rule shown below.

- $\lambda\exists S_1 \in \mathcal{C} : \forall x \in S_1 \rightarrow \exists S_2 \in \mathcal{C}' : \forall y \in S_2 \rightarrow \langle\{x\},\{y\}\rangle \in F(\mathsf{beat})$

In **[7]** we find triples consisting of subsets of $\mathcal{C}_t, \mathcal{C}_t'$ and R_t of $\mathcal{C}, \mathcal{C}'$ and R respectively which satisfy the interpretation rule. In this simple example there is only one such triple which satisfies this distributive interpretation and it is shown below.

- $\mathcal{C}_t = \{\{a,b\}\}$
- $\mathcal{C}_t' = \{\{d,e\},\{f,g\}\}$
- $R_t = \{\langle\{a\},\{d\}\rangle, \langle\{a\},\{e\}\rangle, \langle\{b\},\{f\}\rangle, \langle\{b\},\{g\}\rangle\}$

Any other triple will either not satisfy the interpretation rule or will not be minimal as described in section 5.3.3 of chapter 5. Given that there is only one satisfying triple, $R' = R_t, \mathcal{C}_s = \mathcal{C}_t$ and $\mathcal{C}_o = \mathcal{C}_t'$.

In **[8]**, we create new vertices for each of the verbal argument vertices and in **[9]** we derive the new graph G_4 which as R' is nonempty will have a relational edge between v_s and v_o. This new graph is illustrated in figure 6.9. We have now completed the analysis of the discourse. Given that we have obtained a non-empty graph from the interpretation of the second sentence, we can state that a truthful interpretation has been given, i.e. that the farmers beat the two donkeys they own under the given reading applied.

6.4 Particular Examples

Within the following sections, I will look at how the framework tackles different linguistic examples of pronominal discourse anaphora, including reflexives, quantified donkey sentences, anaphora to sub-sentential information and anaphora to antecedents derived from monotone decreasing quantifiers and explicit negatives.

6.4.1 Reflexive Pronouns

Reflexive pronouns are treated denotationally like any other pronoun. A simple example is shown below.

(274) John loves himself.

The structure of the denotation graph describing the sentence in (274) will be of the form shown in figure 6.14 within a satisfying model. The derived graph contains two vertices, one for *John* and one for the pronoun *himself*. There is one relational edge and one anaphoric edge, between

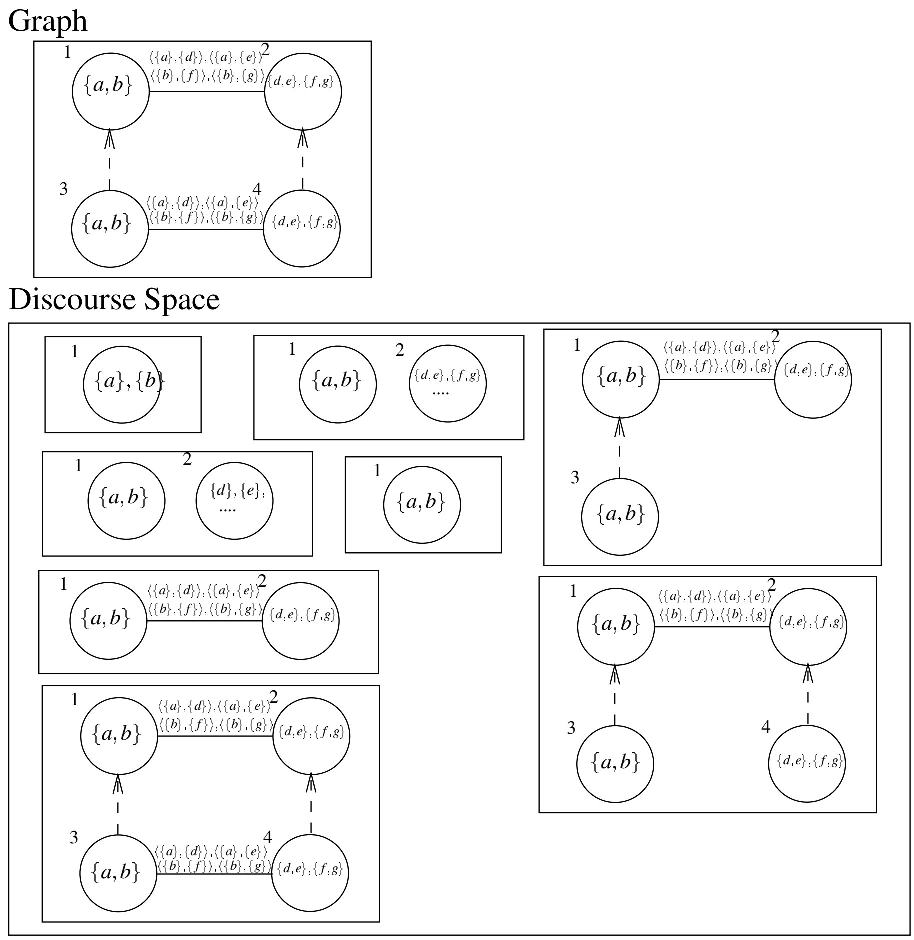


Figure 6.13: Output graph and discourse space after the analysis of the transitive verbal predicate beat in (273).

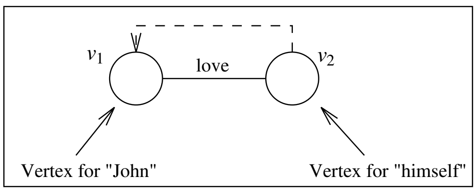


Figure 6.14: Graph describing the sentence in (274).

v_1 and v_2. The denotation of the anaphor in this example would be identical to that of its single antecedent, i.e., the denotation set containing the set with the individual for *John* as the single member. However, in other examples we may wish to create a denotation set for the anaphor which is non-identical to that of its antecedent. Such an example is shown below.

(275) Every farmer loves himself.

The denotation graph derived for (275) in a particular model will be identical in general structure to that for the previous example. However, the denotation for the anaphor is best described by individuating the antecedent vertex for *every farmer* into singleton sets, one for each farmer. The semantic analysis of the bound anaphor-antecedent relation will ensure that each farmer is only allowed to love himself. Other examples can be ambiguous between collective and distributive readings, as shown below.

(276) All farmers love themselves.

Here, if we allow the denotation set for the anaphor to be identical to that of the antecedent (*all farmers*) then under a standard (collective1) collective reading we derive the reading in which *all farmers love all farmers*. If instead we individuate the antecedent and apply a distributive reading we acquire the reading of (275) in which each farmer loves himself. Another collective/distributive ambiguity is shown below.

(277) Some monkeys in the jungle clean themselves.

In (277), we can again either individuate the antecedent for *some monkeys* and obtain via a distributive verbal reading the reading in which each monkey in each collection of *some monkeys* cleans itself, or by allowing the anaphor to be identical to the antecedent and providing a collective verbal reading we can obtain the reading in which each collection of *some monkeys* collectively cleans themselves.

An interesting complex example which involves a reflexive anaphoric reference to sub-sentential information, is shown below.

(278) Most pirates from the Caribbean consider themselves victimized.

On the distributive verbal reading of (278) each pirate considers himself/herself to be victimized. However, there also seems to be a reading in which each pirate considers the collection of pirates in the Caribbean to be victimized. The interesting thing is that the anaphoric reference is referential in nature. Although the syntactic structures are not handled under the basic framework the appropriate graphs under the second reading are shown in figure 6.15. I am assuming that prepositional phrases are treated as relational constructs which in consequence derive a graph edge. In figure 6.15, the graph G_1 would have been derived from the analysis of the nominal phrase *pirates from the Caribbean*. The graph describing the entire sentence in (278) is G_2. The denotation set for the anaphor would be the summation of the sets of pirates from v_1. One could argue that this reference is generic in nature. If so, then GTS shows at worst how the appropriate information for some generic references might be derived even if the framework does not concern itself directly with the interpretation of generic references. However, it is interesting that a variety of anaphoric options are available even within the restricted domain of reflexive pronouns.

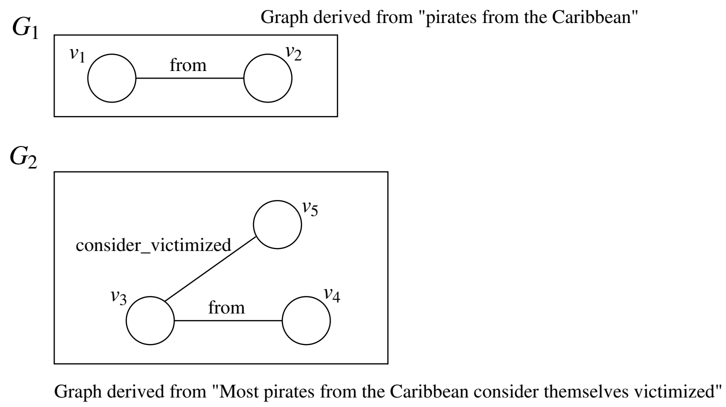


Figure 6.15: Graph describing one reading of the sentence in (278).

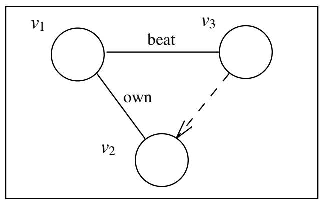


Figure 6.16: Graph describing the quantified donkey sentence in (279).

6.4.2 Quantified Donkey Sentences

I have discussed the analysis of donkey sentences within the framework several times already. It has been shown that the analysis of the quantified donkey sentence in (279) derives a graph of the form shown in figure 6.16.

(279) Every farmer who owns a donkey beats it.

In figure 6.16, vertex v_1 would contain the farmers, while vertices v_2 and v_3 would contain the donkeys, owned and beaten by the farmer. I have shown that both weak (indefinite lazy) and strong (universal) anaphor-antecedent relations can be provided for these sentences. By applying a uniqueness constraint to the object noun phrase for either the *own* or *beat* verbal relations, we can derive the unique antecedent and unique anaphor readings as well. However, there are other discourses similar in nature to that of the quantified donkey sentence which provide interesting situations. For example:

(280) Every farmer who owns a donkey attacks a man who beats it.

The sentence in (280) will derive a graph of the form shown in figure 6.17. The analysis of the verbal predicate beat for the the nominal phrase *man who beats it* will not involve any anaphor-antecedent relation, as illustrated by figure 6.18 which shows the graph constructed in the discourse space from the analysis of the the beat verbal predicate. Only when the attack verbal predicate is analysed will the denotation graph constructed contain anaphoric circuits.

An inter-sentential example is shown below which illustrates the flexibility of the GTS framework.

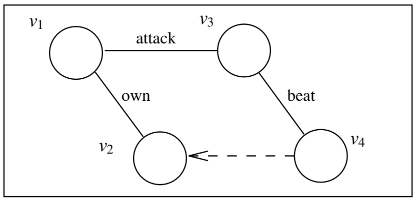


Figure 6.17: The final graph describing the sentence in (280).

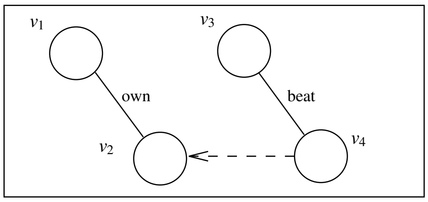


Figure 6.18: Graph derived during the analysis of the sentence in (280) just after the analysis of the beat verbal predicate.

(281) Every farmer owns a donkey. They beat them.

I will assume the first sentence in (281) is given a positive polarity subject and object distributive reading with no uniqueness constraints. In table 6.2 the different implications for the farmers and donkeys are illustrated for several readings in which it is assume that the anaphor *they* refers to all the farmers and the anaphor *them* refers to all the sets of single donkeys owned by the farmers.

One reading for (281) which has not been covered is that where every farmer beats one or more donkeys (not necessarily ones he owns) and every donkey is beaten by a farmer (not necessarily a farmer that owns the donkey). This reading is not covered by the referential reading given above in table 6.2 and seems to require a verbal reading not discussed during the description of the framework in the last two chapters. This reading is the cumulative reading suggested by Scha (1981). Scha originally proposed the reading for transitive verbal relations with numeral quantifiers, an example of which is given below.

(282) 600 Dutch firms have 5000 American computers.

In cumulative reading for (282) can be paraphrased as below.

(283) The number of Dutch firms which have an American computer is 600, and the number of American computers possessed by a Dutch firm is 5000.

An appropriate interpretational rule for this reading is given below, where $\mathcal{C}$ and $\mathcal{C}'$ are the denotation sets for the subject and object arguments and V is the verbal predicate.

(284) $\exists A \in \mathcal{C} : \exists B \in \mathcal{C}' : [\forall x \in A \exists y \in B : \langle \{x\}, \{y\} \rangle \in F(\mathsf{V})] \wedge [\forall y \in B \exists x \in A : \langle \{x\}, \{y\} \rangle \in F(\mathsf{V})]$

This reading can be applied to (281) if the anaphor *they* refers referentially to the set of all farmers and the anaphor *them* refers referentially to the set of all donkeys.

reading	*paraphrase*
Referential anaphor Collective reading	The farmers collectively beat the donkeys they own.
Referential anaphor Distributive reading	Each farmer beats at least one donkey (not necessarily his own).
Bound anaphor Weak anaphoric relation distributive reading	Each farmer beats at least one donkey he owns.
Bound anaphor Strong anaphoric relation distributive reading	Each farmer beats every donkey he owns.

Table 6.2: Some of the various readings attributable to the discourse in (281).

6.4.3 Anaphora to Sub-Sentential Information

The semantic framework derives denotation graphs from sub-sentential phrases. These denotation graphs are retained within the discourse space. All other theories known to the author (including DRT, DPL and DMG) retain only anaphoric information derived from the complete analysis of a discourse. That is, anaphoric information from sub-sentential phrases is not retained after the analysis of a sentence is complete. A question remains whether graphs derived from sub-sentential analysis can be utilized for the purposes of anaphora. In section 6.4.1, the example (278) was shown to have an intra-sentential anaphoric reading which was referential in nature and not bound (like that of the donkey sentences). The example is shown again below.

(285) Most pirates from the Caribbean consider themselves victimized.

The desired antecedent for (285) was *the set of pirates from the Caribbean*. This allowed a reading in which most pirates considered all the pirates from the Caribbean as being victimized. The important consideration here, if one accepts this reading, is that the information from denotation graph for the nominal phrase *pirates from the Caribbean* still has useful content even after the higher level phrase *most pirates from the Caribbean* has been interpreted and a denotation graph derived. It would be even more interesting if inter-sentential references were found to anaphoric information from the analysis of sub-sentential information. An example from chapter 4 is repeated below.

(286) Some farmers who own a donkey beat it. This would not happen if they were inspected by vets.

In (286), there is a reading in which the second sentence can be paraphrased as *the beatings would not happen if vets checked the donkeys owned by farmers*. In particular, this reading does not state that only the donkeys beaten should be checked. The required anaphoric information is only contained within a graph derived from the interpretation of the sub-sentential constituent *farmers who own a donkey*.

Another interesting example is shown below.

(287) Susan saw most clues Bill left. He left them on his desk.

There is a reading (perhaps the preferred reading) in which *them* refers to *all the clues Bill left*. This antecedent is only available from the graph which describes the nominal phrase *clues Bill left*.

Another area, in which the existence of sub-sentential information would seem to be useful is in the analysis of generics. Although generics are not covered by the framework, it would seem that any extension of the framework to incorporate their analysis would find useful the graphs derived from lexical nouns. Some examples are shown below.

(288) John owns a Rottweiler. They are vicious beasts.

(289) The farmers who own a donkey that has one leg and can't hop are unhappy. They are an evolutionary dead end.

In (288), it would seem that the information derived from the graph describing the noun *Rottweiler* will be useful for a generic analysis. Furthermore, (289) shows that we can't just limit the useful information to lexical nouns as the information that seems useful for the analysis of (289) is contained in the graph describing the nominal phrase *donkey that has one leg and can't hop*.

The previous examples seem to suggest that sub-sentential information is useful for purposes other than the correct interpretation of intra-sentential bound anaphors. As most if not all anaphoric semantic theories do not retain information concerning sub-sentential information relating to a sentence after the processing of that sentence, the derivation of the suggested readings in this section would be extremely troublesome.

6.4.4 Monotone Decreasing Quantifiers and Negatives

In the last chapter, it was shown that the analysis of monotone decreasing quantifiers as the negation of monotone increasing quantifiers allowed the correct readings to be given to sentences containing these quantifiers. Furthermore, given their treatment as monotone increasing quantifiers introducing negation, the form of negation must be determined, i.e., either sentential, verb phrase or verbal negation. It was suggested that subject monotone decreasing quantifiers should be provided with verb phrase negation and object monotone decreasing quantifiers with verbal negation. From these stipulations the correct readings could be given to transitive verbal relations involving two monotone decreasing quantifiers or transitive verbal relations with monotone decreasing quantifies interacting with explicit negation. These examples suggest that the analysis of monotone decreasing quantifiers given here has a predictive power not available if they are treated as quantifiers which do not introduce negation into the analysis of verbal relations. Within this section, I will look at how anaphoric reference interacts with monotone decreasing quantifiers and negation.

Webber (1979) coined the terms non-intersective and intersective determiners which correspond to determiners which derive monotone decreasing and monotone increasing (along with non-monotone) quantifiers respectively. She argues that intersective determiners pass on the intersection of the nominal extension with the predicate they are applied to, while non-intersective determiners pass on the extension of the nominal predicate only. She illustrated this with examples such as shown below.

(290) Most linguists smoke, although they know it causes cancer.

(291) Few linguists smoke, since they know it causes cancer.

In (290), the anaphor *they* seems to have a preferred reading in which it refers to *all linguists who smoke*. In (291) however, it is suggested that the preferred reading for the anaphor *they* is to *all linguists*. These readings follow Webber's predictions as *most* is an intersective determiner and *few* is a non-intersective determiner. However, the readings in (290) and (291) may well be biased by the subject matter. A different situation is shown below.

(292) Few students went to the opera, they went to the disco.

(293) Few MPs came to the party, but they had a good time. (Elworthy (1993, p. 132))

In (292), I suggest the preferred reading is that the anaphor *they* refers to the many students who didn't go to the opera. Elworthy (1993) suggests that in (293) the anaphor *they* has a preferred reading in which it refers to the MPs who came to the party. These readings contradict the non-intersective predications of Webber. Furthermore, similar contradictions can be found for intersective determiners.

(294) Most people in France smoke, although they are told repeatedly by the Government that it causes cancer.

In (294), the anaphor *they* has a reading in which it refers to all the people in France, not just the ones that smoke. Indeed, I think one can obtain this non-intersective reading in (290) as well. Indeed, when so called intersective determiners are combined with negation other non-intersective references become available.

(295) Most linguists do not smoke, since they know it causes cancer.

(296) Most MPs did not come to the party but they had a good time.

In (295), there seems to be a reading in which the anaphor *they* refers to all linguists, while in (296) I think it is just possible to obtain a reading in which *they* refers to the MPs who did go to the party.

The connection between these examples becomes more obvious if we treat monotone decreasing quantifiers as monotone increasing quantifiers with negation. Then, the interesting anaphoric references can be understood as being due to the effects of negation. In general then, when negation is involved three types of antecedent for these subject quantifiers become available. Firstly, we have the antecedent derived from the nominal phrase alone, as suggested in (291) and (295). This antecedent is available in the denotation graph derived for the interpretation of the required nominal phrases in GTS. Secondly, we have the intersective cases, which for GTS, covers the examples such as (292) and the example below.

(297) Most people do not drive recklessly, since they know it can be dangerous.

In (297), the anaphor *they* seems to refer to the people who do not derive recklessly. Here, GTS provides the required denotation within the denotation graph derived from the analysis of the transitive verbal predicate, i.e. the analysis of the phrase *most people do not drive recklessly* in (297). The third case is the most complex, and for GTS it occurs in examples such as (293) and (296), repeated below.

(298) Few MPs came to the party, but they had a good time.

(299) Most MPs did not come to the party but they had a good time.

In both, (298) and (296) the correct denotation can be derived in GTS only by subtracting the individuals that satisfy the verbal predicate from those that satisfy the nominal phrase, i.e., in (299) subtracting the set of MPs that don't come to the party from the set of all MPs. The anaphoric information for this reading will be available within the vertices describing the MPs taken from the graph derived from the analysis of the nominal phrase *MPs* and the graph derived from the analysis of the sentence *most MPs did not come to the party*, respectively. However, in my opinion, this last possibility is the hardest of the three antecedent references available in these sentences to grasp.

6.4.5 Miscellaneous Examples

I shall look at within this section several miscellaneous examples not so far discussed. Each has been frequently discussed within the literature as representing a difficult problem for any theory of noun phrase anaphora.

Dekker (1991, p. 89) presents the following examples.

(300) No farmer beats a donkey he owns. He doesn't kick it either.

Dekker shows that the discourse in (300) can not be correctly handled by DMG due to problems concerning treatment of the determiner *no* as constructing either an upward or downward monotone quantifier combined with the problems of deriving either a static or dynamic negation within the dynamic logical framework of DMG. Dekker notes that[5]:

> A plausible interpretation would result if we could take *No farmer beats a donkey he owns* to be equivalent with *Every farmer does not beat every donkey he owns*, with a static negation of the TV *beat*.

However, this is exactly how GTS interprets the monotone decreasing quantifier *no farmers*, i.e., as the negation of a monotone increasing quantifier. Furthermore, as the notion of dynamics as expressed within the dynamic logic frameworks does not occur within GTS, the difficulty of deriving a static or dynamic interpretation of negation does not occur. All anaphoric information is stored within the discourse space and it is up to constraints to limit the accessing of this information.

The denotation graph derived for the discourse in (300) with respect to a satisfying model is given in figure 6.19. The vertex v_1 would contain the set of all farmers who do not beat a donkey they own, while v_3 would contain the set of donkeys not beaten by these farmers. The graph derived from the first sentence in (300) would be the subgraph containing the vertices v_1, v_2, v_3. This graph is utilized by the second sentence in constructing the appropriate reading where the two anaphors *he* (vertex v_4) and *it* (vertex v_5) are treated as bound anaphoric pronouns. The treatment of monotone decreasing quantifiers as monotone increasing quantifiers with negation would provide the correct anaphoric information in v_1 and v_3 while the dynamic effects

[5] Dekker (1991, p. 93).

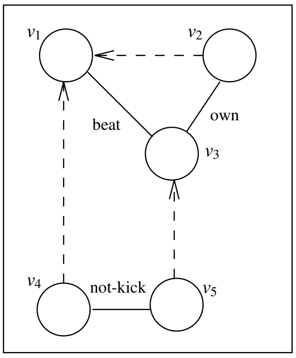


Figure 6.19: The denotation graph for the discourse in (300) within a satisfying model.

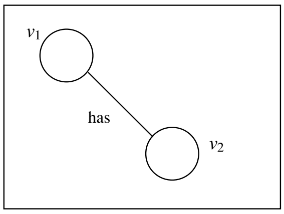


Figure 6.20: The denotation graph for the noun phrase *every man who has a dime* in (301) with respect to a satisfying model.

of anaphoric information are straightforwardly handled by the threading of the discourse space though an interpretation of a discourse.

Chierchia discusses the following well known example.

(301) Every man who has a dime will put it in the meter.

The sentence Chierchia proposes is best read under the unique anaphor reading in which each man may have several dimes but he puts only one in the meter. GTS as it stands does not handle prepositional phrases and, therefore, I shall assume I can treat (301) as containing the compound verbal relation put-in-the-meter. The analysis of the noun phrase *every man who has a dime* will derive, within a satisfying model, a graph of the form shown in figure 6.20. The vertex v_1 would contain the set of men who have a dime and the vertex v_2 would contain the dimes they have. This graph would be extended to form the denotation graph describing the whole sentence. This final graph is shown in figure 6.21. A uniqueness constraint would have been enforced on the

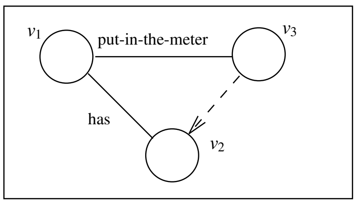


Figure 6.21: The denotation graph for the sentence in (301) with respect to a satisfying model.

object argument to the verbal predicate put-in-the-meter. Therefore, in this graph v_2 and v_3 will contain the dimes (enforced to be only one per man) which the men put in the meter. We have therefore obtained the correct reading for this sentence. However, interesting anaphoric situations can be found if this sentence is extended into a larger discourse, as shown below.

(302) Every man who has a dime will put it in the meter. They use them at the toll-gate, too.

What is interesting about this discourse is that the preferred reading of the second sentence is that the men use the dimes they have at the toll-gate. That is, the bound anaphor *them* needs to reference the information from the denotation graph derived from the noun phrase *every man who has a dime*, i.e., the graph in figure 6.20. The denotation graph derived from the analysis of the first sentence (shown in figure 6.21) does **not** contain the required information as here there is only information concerning the dimes placed in the meter by the men. Any anaphoric theory which only retains the anaphoric information derived from all the constraints in the discourse will not contain the appropriate anaphoric information for the correct analysis of (302). In effect, this discourse reinforces the retention of what I have termed sub-sentential information.

Beaver (1991, p. 149) discusses the following discourse.

(303) Alice is a little girl and anyone who is little can fit through the door. But nobody who has drunk the potion can fit through the door, and she's drunk the potion. She is very confused.

Beaver uses this discourse to illustrate the problems DMG (and possibly DRT) have with contradictory discourses. Both theories tie anaphoric information closely to a truth-conditional analysis. After the analysis of the first two sentences contradictory statements will have been processed and there will be no individuals which satisfy the truth-conditional requirements of these sentences. Thus, when DMG comes to interpret the third sentence, there is no individual that can take the referent of the pronoun *she*. Beaver suggests that DRT on the surface seems to fare better as:

> DRT provides an algorithm for building DRSs that is insensitive to the truth of the discourse. Thus DRT has no problem explaining how *she* gets bound to the discourse marker for *Alice*[6].

However, the entire DRS describing the discourse still can't be embedded in a model and so "we still get no explanation of why we might feel *she* actually refers to Alice the individual in our model"[7] In GTS, however, the denotation graph derived for the proper name *Alice* would be available in the discourse space and available for anaphoric reference. This is irrespective of the fact that the analysis of the second sentence would derive a denotation graph containing only empty vertices. In this respect GTS is impervious to contradictory discourses with respect to the availability of anaphoric information derived from non-contradictory subsections of a discourse.

---

[6]Beaver (1991, p. 150).

[7]Beaver (1991, p. 150.).

Chapter 7

Computational Issues

This chapter will look at the computational issues surrounding any implementation of the GTS semantic framework. The chapter is broken into two halves, in the first half I will discuss in an informal manner the computational complexity of the anaphoric and non-anaphoric sub-parts of the semantic framework. In the remainder of the chapter, I will look briefly at a particular implementation of the GTS framework.

7.1 Computational Complexity

Within this section I will discuss various aspects of the framework with respect to its probable computational complexity. Although no formal proofs of the complexity of different aspects of the framework will be given, I will look at certain sub-problems within the GTS semantic interpretation and show how they are related to problems which have known complexity. I will begin by giving a brief overview of computational complexity theory before continuing to look at the GTS framework itself. I will utilize the text by Garey and Johnson (1979) as the basis for this introduction to complexity theory.

The two main components of computational complexity theory are *problems* and *algorithms* for those problems. Garey and Johnson (1979, p. 4) describe a *problem* and an *algorithm* as:

> ...a problem will be a general question to be answered, usually possessing several *parameters*,... A problem is described by giving: (1) a general description of all its parameters, and (2) a statement of what properties the answer, or *solution*, is required to satisfy. An *instance* of a problem is obtained by specifying particular values for all the problem parameters.
>
> *Algorithms* are general, step-by-step procedures for solving problems

It is traditional in complexity theory to attempt to describe a problem in terms of a decision problem, i.e., a problem in which the answer is to be either "yes" or "no".

In general, computational complexity theory is concerned with determining how "efficiently" certain problems can be solved. Most often, efficiency is equated with time, i.e. we are concerned with finding out how fast certain problems can be solved. As it is algorithms that solve problems, we are concerned with determining the efficiency of certain algorithms for particular problems, or more interestingly, the efficiency of the fastest possible algorithm to solve a particular problem. Usually, the time requirements of an algorithm are expressed with respect to the "size" of the particular problem instance in question, where the size of a problem is the amount of input data needed to describe that problem. That is, it is assumed that the time taken for a particular algorithm to solve a particular problem instance will vary with respect to the size of that problem instance. The input data for a problem will be described via an *encoding scheme*. Some possible encoding schemes might waste space and artificially lengthen the input data size. Thus in general we require a *reasonable* encoding scheme. A *reasonable* encoding scheme is not a well-defined concept although Garey and Johnson (1979) suggest that it is any scheme which is concise and not padded with unnecessary information or symbols and which is expressed in any fixed base other than 1.

The time complexity of an algorithm for a problem can be expressed as some function of the (encoded) input data size for that problem. Two important function types are polynomial and exponential. They are described by Garey and Johnson (1979, p. 6) as follows:

> Let us say that a function $f(n)$ is $O(g(n))$ whenever there exists a constant c such that $|f(n)| \leq c.|g(n)|$ for all values of $n \geq 0$. A *polynomial time algorithm* is defined to be one whose time complexity function is $O(p(n))$ for some polynomial p, where n is used to denote the input length. Any algorithm whose time complexity function cannot be so bounded is called an *exponential time algorithm*.

A problem that can be solved by an algorithm with polynomial time complexity is thought of as *tractable*. If no polynomial time complexity algorithm can solve a problem then it is an *intractable* problem.

An important class of problems is the class of NP-complete problems. Cook (1971) was the first person to define and find an NP-complete problem. He used the technique of reducing one problem into another by defining a transformation that maps any instance of the first problem into an equivalent instance of the second. This technique allows an algorithm which solves the first problem to be converted into an algorithm which solves the second. Cook looked particularly at polynomial time reducibility. That is, reductions which can be accomplished by a polynomial time algorithm. If one has a polynomial reduction from one problem to another and a polynomial algorithm to solve the second problem, then one can construct a polynomial time algorithm to solve the first. Cook then concentrated on the class NP of decision problems that can be solved in polynomial time by a nondeterministic computer[1]. He showed that one particular problem, Satisfiability, had the property that every other problem in NP can be polynomially reduced to it. Thus, if Satisfiability could be solved by a polynomial time algorithm so could all problems in NP. Furthermore, if any problem in NP was intractable then so would Satisfiability be intractable. Satisfiability is in some sense the "hardest" problem in NP. Other problems in NP which also satisfy these properties have also been found and the class as a whole is called the class of NP-complete problems. These problems are distinguished by the fact that no polynomial time algorithm has been found for them, but neither have they been proved to be intractable. Furthermore, if any polynomial time algorithm were found for any of them then they would all be solvable in polynomial time and if any of them were found to be intractable then they would all

[1]The description of a nondeterministic computer can be found in Garey and Johnson (1979, pp. 27-32).

be intractable. The problem of deciding whether NP-complete problems are intractable or not is one of the outstanding open problems in computer science.

I will now consider what would be required to determine the complexity of GTS as a whole. In order to accomplish this I will need to describe a decision problem utilizing the GTS framework. This problem will make use of the interpretation function $[\![]\!]$ whose specification is repeated below, where α is a semantic representation, M is a model, I is a set of identifiers, $\mathcal{CONS}$ is an anaphoric constraint function, $\mathcal{RES}$ is an anaphoric resolution function, G and G' are denotation graphs, $\mathcal{D}$ and $\mathcal{D}$' are discourse spaces and i is an identifier.

(304) $[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$

The decision problem I shall derive will be to determine whether the semantic interpretation of the semantic representation for a declarative sentence derives a truthful or false interpretation. The determination of truth in GTS is repeated below.

- Given a sentence S whose semantic representation is the feature structure α, α is *true* with respect to a model M, set of identifiers I, a semantic interpretation function $\mathcal{CONS}$, a semantic resolution function $\mathcal{RES}$, and discourse space $\mathcal{D}$,
  if $[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(\langle\{\},\{\},\{\}\rangle,\mathcal{D}),(i,G,\mathcal{D}')\rangle$ and $\exists v \in G : v \neq \langle j,\{\}\rangle$ for some j.

A decision problem can be described by a *generic* instance specifying the structures used in the problem and a question asked about those structures.

- *Instance*:
  1. A declarative sentence whose semantic representation is α.
  2. A model $M = \langle D,F\rangle$, where F only provides information concerning predicates given in α, while D is the set of individuals covered by F.
  3. A set of identifiers I, where $|I|$ is equal to the number of nominal predicates, proper name predicates and pronominal predicates specified in α.
  4. An anaphoric constraint function $\mathcal{CONS}$.
  5. An anaphoric resolution function $\mathcal{RES}$.
- *Problem*: Is the interpretation $[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle(\langle\{\},\{\},\{\}\rangle,\{\}),(i,G,\mathcal{D})\rangle$ true or false.

This problem concerns only single declarative sentences. A similar problem could be defined for discourses: the problem being whether the last sentence in the discourse determined a truthful or false interpretation.

In general, a particular reasonable encoding scheme will need to be devised to transform the information in the above problem into a string. Due to the nature of a reasonable encoding scheme both the problem and the encoding should not contain redundant information. That is, we should not allow models M or sets of identifiers I which contain redundant information as they may artificially lengthen the input data of the problem and effect the resulting complexity analysis. Under a reasonable encoding the length of the input string describing any instance of the problem described above would only be effected by some reasonable factors such as the size of the semantic representation and the size of the restricted model domain[2]

[2]I am assuming only finite models here.

As has been mentioned, I will not be trying to derive a complexity result for the above GTS problem, and indeed such a result may prove to be quite difficult. However, any attack on the problem would probably be via a component proof which looks at analysing sub-components of the problem and combining them together to arrive at an overall complexity proof. Under such a proof, one would probably associate with each component the interpretation given to a particular type of predicate. The GTS interpretation as it stands provides a compositional analysis and this suggests that a component complexity analysis could be provided in a similar compositional manner.

In the next few sections I shall look at several pertinent parts of the framework which would contribute to any component complexity analysis of GTS. Under the viewpoint that a component analysis would associate components with types of predicates the different sub-problems within the GTS interpretation are shown below.

- Verbal Predicates.
  - Determining verbal relations.
  - Finding graph circuits.
  - Solving the constraint satisfaction problem for denotation graphs.
- Determiner Predicates.
  - Determining generalized quantifier denotations.
- Nominal Predicates.
  - Determining lexical noun denotations.
- Pronominal Predicates.
  - Determining anaphor denotations.

I will concentrate in sections 7.1.1 and 7.1.2 on the graphical problems given above, that is, finding circuits within a graph and solving the constraint satisfaction problem for denotation graphs. The remaining problems fall into two areas. There are the more standard though not necessarily tractable problems: determining the denotations for lexical nouns (i.e. from the model function F); determining the denotation for generalized quantifiers (i.e. the witness sets that satisfy the quantifier); determining a verbal relation (i.e. the sets from the arguments to the verbal relation which satisfy the particular verbal reading required). Then there is the problem of determining the denotation for an anaphor. Under GTS, this centrally concerns the functions $\mathcal{CONS}$ and $\mathcal{RES}$. How these functions should be handled in a computational complexity analysis is uncertain. These functions are given by the problem instance and they can be utilized in a purely extensional manner as part of the input data encoding. As these functions carry out all the work of deciding upon anaphor and antecedent denotations, this means that the computational complexity of an anaphor predicate depends crucially on the complexity of "accessing" the required information from these anaphoric functions.

On a final note, even if the complexity of GTS can be derived it could not be used as evidence for the complexity of semantic processing of anaphoric discourses in general. That is, we should

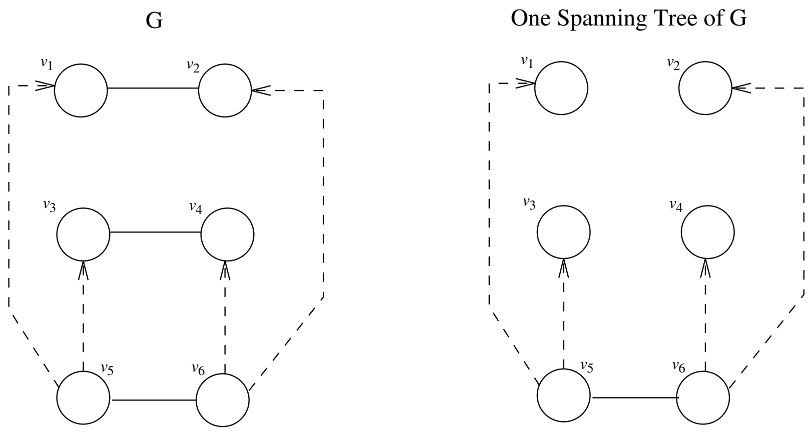


Figure 7.1: A graph with associated spanning tree

be especially wary of carrying over results about a particular semantic framework to that of the general linguistic problems the semantic framework addresses. Hopefully, the two would not be totally unrelated but a great deal of extra evidence would need to be given before such claims could be made. However, the complexity of the framework would provide an upper bound on the complexity of the particular problem it solves providing at least a mark to aim at for subsequent theories.

7.1.1 Finding Circuits in a Graph

In order to provide a consistent labelling to a denotation graph, the anaphoric circuits within that graph need to be found. This can be accomplished by utilizing some standard techniques from graph theory[3]. The distinct circuits in a graph can be derived by finding a spanning tree for the graph. A spanning tree of a connected graph G is a sub-graph which is both a tree and which contains all the vertices of G. As an example, figure 7.1 shows an example denotation graph and one possible spanning tree derived from it. The edges not contained in the spanning tree can be utilized to determine the *fundamental* circuits for the graph G. To determine the fundamental circuits, for each edge not in the spanning tree (i.e., each edge in the so-called co-tree), find the (single) path through the spanning tree connecting its vertices. In figure 7.1, the relational edge between v_1 and v_2 is not in the spanning tree. The path between its vertices is v_1, v_5, v_6, v_2. If the edge between v_1 and v_2 is added we obtain a fundamental circuit. A set of fundamental circuits with respect to a spanning tree of a graph can be used to derive the set of all possible circuits for a graph via appropriate application of the ring sum graph operation on members of the set of fundamental circuits[4]. From this set of circuits we must reject any which utilize two anaphoric edges from the same anaphoric edge set, i.e., we reject any circuit which utilizes more than one anaphoric edge emanating from a single anaphor vertex. The remaining circuits are anaphoric circuits.

The complexity of finding a spanning tree in a graph is O(max($n, |E|$)) where n is the number

[3]Gibbons (1985) provides a good introduction to algorithmic graph theory.

[4]See Gibbons (1985, pp. 54-56).

of vertices and E is the number of edges in the graph. The worst case complexity for finding all the fundamental circuits of a graph is $O(n^3)$. These complexity results can be found in Gibbons (1985).

7.1.2 Denotation Graphs as Constraint Networks

During the semantic interpretation denotation graphs are interpreted as constraint networks for which a consistent labelling needs to be found. Denotation graphs treated as constraint networks are utilized within GTS to derive anaphorically acceptable verbal readings and also to derive maximally consistent denotation graphs after one or more vertices in a graph have been modified and we wish to conform the other vertices in the graph to respect these changes. The labelling constraints used to determine a consistent labelling are given again below.

- Relational Edge Constraint.
  Given a relational edge $\langle v, v', R\rangle$ where the labels for v and v' are S and S', respectively, then it must be that $\langle S, S'\rangle \in R$.

- Anaphoric Edge Constraint.
  Let $\mathcal{M}$ be the set of sets of maximal non-conflicting anaphoric circuits for G. If $\mathcal{M}$ is not empty then there must be some set of circuits $m \in \mathcal{M}$ such that the label on each anaphor vertex identified in m is identical to the label of its activated antecedent identified in m.

A particular labelling of a graph must satisfy the above constraints to be a consistent labelling. The problem of finding a consistent labelling for the constraint network derived from a denotation graph is closely related to the general area of finite constraint satisfaction problems (CSPs). Cooper, Cohen and Jeavons (1994) describe a finite constraint satisfaction problem (with binary constraints) as follows:

> A *finite constraint satisfaction problem* (CSP) consists of a finite set of nodes, N (identified by the natural numbers $1, 2, \ldots, n$), each of which has an associated finite set of possible labels A_i. The labellings allowed for specified pairs of nodes are restricted by a set of constraints, C. Each constraint $C_{ij} \in C$ is a list of pairs of labels from A_i and A_j which may be simultaneously assigned to the nodes i and j, i.e., $C_{ij} \subseteq A_i \times A_j$. A solution to a CSP is a labelling of the nodes which is consistent with all the constraints.

Denotation graphs treated as constraint networks closely resemble the above problem. The vertices of a graph correspond to the nodes of a CSP. The subsets of a set within a denotation set described by a vertex correspond to the labels of a node. A relational edge $\langle v_i, v_j, R\rangle$ corresponds to a constraint C_{ij} between nodes i and j where $C_{ij} = R$. Anaphoric edges and their associated constraints don't easily map onto the above notion of a finite constraint satisfaction problem. They can, however, be incorporated into the CSP description as n-ary constraints.

The anaphoric edge constraint requires that sets of maximal non-conflicting circuits be found. This implies that there may be sets of circuits from a graph which conflict with each other, i.e, they utilize distinct anaphoric edges from some anaphor vertex. We can group together these

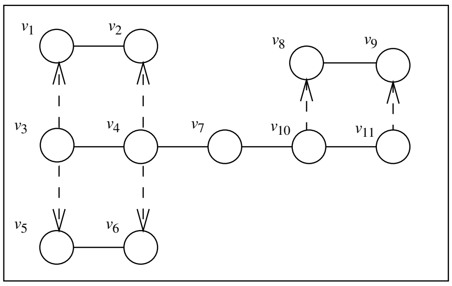


Figure 7.2: An example denotation graph.

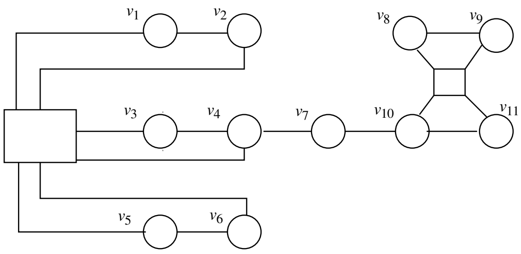


Figure 7.3: Constraint network for denotation graph in figure 7.2

conflicting circuits into sets. Viewed in this manner each group of conflicting circuits defines an n-ary constraint over the vertices involved in the circuits they describe.

An example denotation graph is shown in figure 7.2. This denotation graph has 3 circuits, $\mathcal{A}_1 = v_1, v_2, v_4, v_3, v_1$, $\mathcal{A}_2 = v_3, v_4, v_6, v_5, v_3$ and $\mathcal{A}_3 = v_8, v_9, v_{11}, v_{10}$. The circuits $\mathcal{A}_1$ and $\mathcal{A}_2$ form a group of conflicting circuits, while the circuit $\mathcal{A}_3$ forms a group of its own. If we convert this denotation graph into a constraint network we would obtain a CSP with the form shown in figure 7.3. The square boxes in this figure represent n-ary constraints. Each one of these n-ary constraints will describe the possible constraints on the nodes derived from picking each possible set of non-conflicting maximal circuits from the set of conflicting circuits. In this case, for the 6-ary constraint defined over the vertices $v_1 - v_6$ the constraint should describe the fact that either circuit $\mathcal{A}_1$ could be "chosen" in which case the label for v_1 must equal the label for v_3 and the label for v_2 must equal the label for v_4 with the labels for v_5 and v_6 not restricted (by the anaphoric constraint) or the circuit $\mathcal{A}_2$ could be "chosen" in which case the label for v_3 must equal the label for v_5 and the label for v_4 must equal the label for v_6 with the labels for v_1 and v_2 not restricted (by the anaphoric constraint). For the 4-ary constraint defined over the vertices $v_8 - v_{11}$ the situation is much simpler as there is only one circuit here and thus the constraint must specify that the label for v_8 must equal the label for v_{10} and the label for v_9 must equal the label for v_{11}.

Finding all solutions (labellings) to finite constraint satisfaction problems for arbitrary networks is NP-complete (Haralick *et al.*, 1978, p. 206), even for CSPs with only binary constraints. Finding a single solution is also NP-complete. In the remainder of this section I will look at results showing how the combinatorial explosion of CSPs can be mitigated in certain situations, before finishing by looking at how these results might concern an implementation of GTS.

Labelling all the nodes in all possible ways in order to find those labellings which satisfy all the constraints in a CSP (the so-called generate and test algorithm) will always be exponential in the best, average and worst case. The most common general purpose algorithm for solving CSPs is the backtracking algorithm which is linear in the best case but is still exponential in the average and worst case. Due to inefficiencies in the backtracking algorithm several so-called consistency algorithms have been devised which can make local transformations on a CSP which allow a subsequent utilization of the backtracking algorithm to have a better chance of providing an efficient solution to the, now transformed but still equivalent, CSP. Mackworth (1977) devised the notion of arc-consistency. A CSP is arc-consistent if for any two nodes i and j that are connected via a binary constraint, C_{ij}: for every label in i there is a label in j which satisfies C_{ij}. More formally, using the terminology described above, for all nodes i and j, $\forall x \in A_i : C_{ij} \in C \rightarrow \exists y \in A_j \langle x, y \rangle \in C_{ij}$. Mackworth and Freuder (1985) discuss an arc-consistency algorithm with worst case time complexity O(ea^3) where e is the number of edges and a is the number of labels in the network. Mohr and Henderson (1986) derive an optimal arc-consistency algorithm whose time complexity is O(ea^2). Montanari (1974) developed the notion of path-consistency. A CSP is path consistent if for every $C_{ij} \in C$ for any labels $x \in A_i$ and $y \in A_j$ such that $\langle x, y \rangle \in C_{ij}$ for every path between i and j labels can be found to satisfy the constraints between adjacent nodes on the path. Mohr and Henderson (1986) provide the basis of a path-consistency algorithm which has worst case time complexity O(n^3a^3), where n is the number of nodes and a is the number of labels in the network. Han and Lee (1988) correct an error in Mohr and Henderson's original algorithm to produce a revised path consistency algorithm which has the same worst case time complexity. Freuder (1978) has generalized the notion of consistency to k-consistency, where arc consistency and path consistency correspond to 2-consistency and 3-consistency, respectively. Cooper (1989) provides an optimal k-consistency algorithm. His k-consistency algorithm has, as expected, exponential time complexity in the worst case.

Consistency techniques try to find local solutions which might reduce the time taken to find global solutions to a CSP. However, this work is based on the assumption of being applicable to general constraint satisfaction problems over arbitrary networks. Other work has looked at finding restricted classes of the general constraint satisfaction problem which allow efficient solution algorithms. In general, there have been two types of restrictions placed on CSPs, in particular, the structure of the network described by a CSP and the type of constraints that are allowed within a CSP. Montanari and Rossi (1991) have looked at graphs that can be generated by a kind of context-free (hyper)graph grammar and have shown that networks based on these graph structures can be solved in time linear with the number of edges. Freuder (1982) defines some basic concepts which help determine graph structure, as given below.

Definition 25: *An ordered constraint graph* is a constraint graph in which the nodes are linearly ordered to reflect the sequence of label assignments executed by a backtrack search algorithm. The *width of a node* is the number of arcs that lead from that node to previous nodes, the *width of an ordering* is the maximum width of all nodes, and the *width of a graph* is the minimum width of all orderings of that graph.

From these definitions he shows that if a constraint graph has width 1 (i.e., it is a tree) and if it is arc-consistent, then it admits backtrack-free solutions. Also, if the width of a constraint graph is 2 and it is path-consistent, then it admits backtrack-free solutions. Dechter and Pearl (1988) extend these techniques to provide a heuristic based approach to finding backtrack-free solutions. Hentenryck, Deville and Teng (1992) devise an arc consistency algorithm which can be applied to networks containing functional, anti-functional and monotonic constraints with time complexity

O(el) where e is the number of edges and l is the number of labels on the node which has the highest number of labels in the network. Cooper, Cohen and Jeavons (1994) show that CSPs containing only so-called 0/1/all constraints can allow backtrack-free solutions after application of a path-consistency algorithm.

These results have implications for any computational implementation of GTS. From the discussion at the start of this section I have shown the close similarity between CSPs and finding a consistent labelling for a graph in GTS. This would suggest that a polynomial transformation between the finite constraint satisfaction problem and the labelling problem in GTS is possible, assuming that the construction of the appropriate n-ary constraints can be accomplished efficiently. Furthermore, given that CSPs are NP-complete and that for reasons similar to those for CSPs the labelling problem for denotation graphs is NP-easy[5], it would seem that the labelling problem for graphs in GTS is NP-complete also. The consistency algorithms given above may well help to mitigate the combinatorial explosion for any implementation of GTS. However, this assumes that arbitrary denotation graphs can be derived from the GTS analysis of natural language. Haddock (1988; 1992) has shown that for non-anaphoric discourse the CSPs derived are all tree-like and can be solved by arc-consistency techniques. Haddock notes that some simple instances of intra-sentential anaphora derive CSPs with width 2. With regards to GTS, width measures are not generally applicable as anaphoric constraints are treated as n-ary constraints and width measures only apply to binary CSPs. However, arbitrary n-ary constraints are only required for anaphors referencing multiple antecedents. For singular references it would be possible to treat all anaphoric edges as binary identity constraints relating identical labels in the anaphor and antecedent vertices.

I will show that all the denotation graphs derived by GTS when anaphoric references are restricted to single antecedents only are width 2 denotation graphs. In order to show this I will first provide a proof of the following proposition.

Proposition A: Any binary CSP can be translated into a width 2 binary CSP.

Proof: Given a binary CSP N. Find a spanning tree of the constraint network described by N. A tree always has width 1 (Freuder, 1982). Order the nodes to obtain a width 1 ordering. For any edges between nodes i, j in the co-tree (i.e., edges not in the spanning tree) connect them via a new node k, where $A_k = A_j$ (i.e., nodes k and j have the same domain). Let $C_{i,k} = C_{i,j}$ and $C_{k,j} = \{\langle x,x\rangle | x \in A_k \wedge x \in A_j\}$ (i.e., the identity constraint). Let node k be the k^{th} in the ordering. The node k will have width 2 as it connects nodes i and j which have been ordered from the spanning tree. As, all edges in the co-tree can be added via the addition of an extra node and are certain to be of width 2, the ordering of the network as a whole is width 2. The solutions for the original CSP can be found by finding the solutions to the transformed CSP and ignoring the labels for the additional nodes of the transformed CSP network. □

Next, I will illustrate a fact concerning the structure of denotation graphs derived in non-anaphoric discourse.

Proposition B: A GTS framework without anaphor predicates can derive only denotation graphs which are tree structured i.e., they contain no circuits.

Proof: Without anaphor edges, the only edges connecting vertices in a denotation graph can be relational edges. Relational edges are derived during the processing of transitive verbal re-

[5] One could non-deterministically present a labelling of all the vertices of a graph and then check in polynomial time whether all the constraints satisfied this labelling.

lations. The interpretation of arguments to a transitive verbal predicate derive a graph at least containing two vertices over which the verbal relation will be derived. For circuits to be derived there must be a path between the argument vertices. However, each argument to a verbal predicate describes a separate semantic representation which will derive separate graph components. Therefore, no path between argument vertices to a transitive verbal relation is possible and the resulting graphs are trees (branching is derived through the interpretation of relative clauses). □

The original proposition can now be proved.

Proposition C: Denotation graphs derived from GTS in which anaphors refer only to single antecedents are all of width 2.

Proof: By the proof of proposition B non-anaphoric denotation graphs are tree structured. However, denotation graphs derived from anaphoric discourse contain circuits. As these graphs differ from the former only by the appearance of anaphoric edges it must be that anaphoric edges derived through an interpretation induce circuits. From the proof of proposition A edges in the co-tree were added to the CSP via a new "buffer" node and identity constraint. However, this maps exactly the interpretation of anaphors in GTS, where anaphor vertices act like the "buffer" node and the anaphoric edge can be seen as an identity constraint. As these CSP networks are of width 2, so must the denotation graphs. □

Unfortunately, (Dechter & Pearl, 1988, p. 10) state that (all known) path-consistency algorithms sometimes add extra edges to a constraint network which can result in the width of the derived network being greater than 2. That is, although Freuder has shown that a width-2 network which is path-consistent has backtrack-free solutions, not all width-2 networks which are made path-consistent remain width-2 after the application of a path-consistency algorithm. Indeed, as the proof of proposition A shows that all binary CSPs can be made width 2, if this transformation can be achieved in polynomial time (which seems likely) and if path-consistency algorithms never increased the width of a network then all binary CSPs could be solved in polynomial time, which in turn would prove that the labelling problem for CSPs, an NP-complete problem, could be solved in polynomial time[6].

The work described previously on the tractability of CSPs with restricted constraints seems less applicable as these constraints seem too restrictive to cover the types of constraint derived from natural language relations. Cooper, Cohen and Jeavons (1994) investigate 0/1/all constraints which cover constraints C_{ij} where each label $x \in A_i$ is consistent with zero, one or all labels in A_j. There seems no evidence to suppose natural language is restricted to these type of relations. Hentenryck, Deville and Teng's (1992) work on monotonic constraints may well be applicable. However, although they show that arc-consistency can be derived in linear time for monotonic constraints, an arc-consistent network may not be minimal and may well still require exponential time to solve. The work by Montanari and Rossi on hyper(graph) grammars may well be applicable. However, the determination of whether the graphs derived by GTS fall within the coverage of the types of (hyper)graph grammar investigated by Montanari and Rossi is outside the scope of the present work.

[6] Dechter and Pearl (1988, p. 10) state that they are "unsure as to whether a width-2 ordering suffices to preclude exponential complexity". The proof of proposition A, which to my knowledge has not appeared before in the CSP literature, shows that this question is dependent on polynomial time algorithms being found for NP-complete problems and thus given that this seems to be unlikely, it would seem that width-2 networks do not preclude exponential complexity.

Another area of constraint satisfaction research that has not so far been discussed is that of dynamic constraint satisfaction. The previous discussion of constraint satisfaction has assumed a static network with static constraints. However, within GTS a denotation graph is incrementally constructed through the interpretation of a sentence. It would be wasteful to ignore solutions to labellings of previous graphs in the analysis and begin from scratch the labelling of a graph which is an extension of a previous graph. In general, dynamic constraint satisfaction problems involve the incremental addition or removal of labels or constraints from a given static CSP. That is, a dynamic CSP is a sequence of static CSPs $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_n$, where $\mathcal{P}_i (1 \leq i \leq n)$ is derived by modifying $\mathcal{P}_{i-1}$. Usually, each modification will be of a constrained nature, e.g., the addition of a new constraint or the reduction of some node's domain of labels. If labels are added or constraints removed the new CSP is a *relaxation* of the former. If labels are removed or constraints added the new CSP is a *restriction* of the former. Relaxations are harder to cope with as they require former rejected labels to be reintroduced. Consistency algorithms for static CSPs do not need to retain information concerning the reasons why a label was rejected. Bessiere (1991) describes a modified static arc consistency algorithm which handles dynamic CSPs. The algorithm retains information as to why labels were previously rejected and thus can handle cases of dynamic relaxation. Dynamic restriction, however, does not require radical extensions to existing static algorithms as solutions already rejected will remain rejected. Through the analysis of a sentence by GTS, each derived denotation graph is a restriction of the previous one in the analysis and thus the added complication of relaxation does not arise.

This section has looked further into the treatment of denotation graphs as constraint networks. I have looked at the various algorithmic possibilities open for helping to solve CSPs in an efficient manner. I have also looked at the various restrictions that can be placed on the class of CSPs in order to obtain efficient sub-classes of CSPs. Finally, I looked at dynamic CSPs, a form of CSP more close to the utilization of constraint networks in GTS. As far as can be determined by the present study the type of CSPs derived in GTS do not fall into an efficient sub-class.

7.2 Implementation

An implementation of the GTS framework has been completed utilizing the language Prolog. The feature-based semantic representations are described within PATR. The implementation utilizes the Mini-PATR system which implements a subset of PATR-II. A left corner parser is provided with the Mini-PATR system and is utilized within the implementation to provide the syntactic parsing of sentences. A grammar has been written following the lines discussed in section 4.3 of chapter 4 and is given in full in appendix B. The grammar allows basic declarative sentences with transitive verbs, simple relative clauses, and negation of the form *does not V* where *V* is some transitive verb. The particular model against which an interpretation is provided is given in a file which can be loaded by the program. A simple example model is shown in figure 7.4. The model domain, *D*, is not explicitly given. The function *F* which provides semantic values to the non-logical predicates used in GTS is given explicitly as a set of pairs. The first item in each pair is the predicate name and the second item the value provided by the model. For example: the predicate donkey is given the value the Prolog list `[d1, d2, d3]`, which can be taken to describe the set $\{d_1, d_2, d_3\}$; the predicate beat is given as its value the Prolog list `[[[f1], [d1]], [[f2], [d2]], [[f3], [d2]] ]` which can be taken to describe

```
[donkey, [d1, d2, d3]].
[farmer, [f1, f2, f3]].
[lawyer, [l1, l2, l3]].
[visit, [ [[l1],[f1]], [[l2],[f2]], [[l3],[f3]] ] ].
[own, [ [[f1], [d1]], [[f2], [d2]], [[f3], [d2]],
        [[f3], [d3]] ] ].
[beat, [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]] ] ].
[hate, [[[f1], [f1]], [[f2], [f2]], [[f3], [f3]] ] ].
[bite, [[[d1], [f1]], [[d2], [f2]], [[d3], [f3]],
        [[d1], [l1]], [[d2], [l2]], [[d3], [l2]] ] ].
```

Figure 7.4: A simple example model.

the set of pairs $\{\langle\{f_1\},\{d_1\}\rangle,\langle\{f_2\},\{d_2\}\rangle,\langle\{f_3\},\{d_2\}\rangle\}$.

Within the following two sections I shall describe the implementation of the two most complex parts of the framework: determining a consistent labelling of a denotation graph, and determining and applying a particular verbal reading. In the final section I shall provide an example run of the implementation.

7.2.1 Graph-Theoretic Analysis

The determination of a consistent labelling for a denotation graph takes account of the fact that the derivation and utilization of constraint networks in GTS is a dynamic process. That is, a constraint network is constructed incrementally through the semantic interpretation of a sentence. Furthermore, each denotation graph derived must describe a minimal constraint network. That is, the vertices of any derived denotation graph describe just those sets of individuals which satisfy the constraints of the interpretation up to that point. For example, the semantic interpretation of the noun phrase *every farmer who owns a donkey* can only be provided after we have determined just which farmers own a donkey from the interpretation of *farmer who owns a donkey*. This places a restriction on how constraint networks are to be treated within the implementation. It is not possible to wait until one has derived the final denotation graph for the sentence being considered and *then* solve the constraint network it describes. However, because each denotation graph is an extension of a previous one, it would be inefficient to solve each constraint network derived from each denotation graph from scratch. Instead, the implementation retains the set of all consistent labellings for a denotation graph. These labellings, can be utilized in determining any consistent labelling of an extended graph. The implementation does not explicitly provide the consistent labels for any vertices not connected by relational edges. The possible labels for such vertices are very unconstrained and if enumerated explicitly would greatly increase the number of consistent labellings. Instead, a Prolog variable is used to represent the *unconstrained label*. These implicit consistent labellings can always be expanded at a later date when the unconstrained vertex is constrained. For example, the Prolog structure for a particular consistent labelling containing an unconstrained vertex is shown below.

(305) `[ l(1,[f1]), l(2,_3), l(3,[d1]) ]`

The structure `l(1,[f1])` states that the label for vertex 1 is `[f1]`, while the structure `l(2,_3)` states that vertex 2 is unconstrained. If after some further interpretation it is found that vertex 2 can take the values `[d1]` and `[d2]` only, and these values are consistent with the above consistent labelling, then we can easily use Prolog unification mechanisms to derive the following two consistent labellings.

(306)
```
[ l(1,[f1]), l(2,[d1]), l(3,[d1]) ],
[ l(1,[f1]), l(2,[d2]), l(3,[d1]) ]
```

In general, the discourse space is treated simply as a list of graphs. Graphs are Prolog structures `gr(N,V,E,A,L)` where `N` is a unique graph number, `V` is a list of vertices, `E` is a list of relational edges, `A` is a list of anaphoric edges and `L` is a list of consistent labellings. Graph numbers are used to simplify and speed up the access of graphs from the discourse space. To save time, the discourse space is not enforced to be a set as required by GTS.

7.2.2 Verbal Readings

The analysis of verbal readings for transitive verbal predicates is carried out in two stages.

1. The appropriate verbal rule is determined from the semantic feature structure of the transitive verbal predicate.
2. The sets of individuals satisfying the verbal rule are derived.

The appropriate verbal rule is derived by checking that the control feature for the verbal predicate has certain values. One such check is shown below.

- ```
  get_so_sec(s1,C,lambda(P,
             exists(a,in(subj),forall(x,singletons(a)
                                ,arg(P,x))))):-
     feat_check(reading:distributive,C),
     feat_check(uniq:no,C).
  ```

The predicate `get_so_sec` determines the appropriate rule for the subject argument to the verbal predicate. The variable `C` contains the features within the ⟨control subject⟩ complex feature. The predicate `feat_check` checks whether the feature of its first argument is contained with the list of features in its second argument. Features are held as infix structures `F:V`, where `F` is a feature and `V` is its value. The rule derived in the above example is shown in (307) which corresponds to the rule in GTS shown in (308).

(307) `lambda(P,exists(a,in(subj),forall(x,singletons(a),arg(P,x))))`

(308) $\lambda P, C_1 . \exists S_1 \in C_1 : \forall S_2 \subseteq S_1 : |S_2| = 1 \rightarrow P(S_2)$

The structures of the form `lambda(V,Form)` correspond to a lambda variable `V` associated with the formula `Form`. Structures `arg(P,x)` correspond to lambda abstracted predicates of the form $P(x)$. Quantified logical variables are translated as Prolog atoms. The subject denotation set, C_1 in (308), is translated as a Prolog atom `subj` in (307). This simplification is allowed
  ```

by the non-coordinate structures covered in the basic framework whereby each transitive verbal predicate will have exactly one subject and object argument.

The appropriate rules for the object and predicate parts of a verbal predicate are similarly derived. The resulting rules are then combined and lambda reduced. After this negations are pushed inwards, transforming the derived rule into one in which negation only has scope over atomic elements. This last procedure is carried out to simplify the subsequent predicates which utilize the verbal rule to determine the validating sets from the arguments to the verbal relation.

Once the appropriate verbal rule is obtained it is processed with respect to the denotation sets for the subject and object arguments to the verbal predicate. This is carried out mainly via a recursive predicate `apply` which has the following form.

- `apply(Form,Assign,Truth,Vals)`

The variable `Form` holds the present formula being considered, `Assign` holds the assignments to variables from the previous analysis of the verbal rule, `Truth` takes a boolean value and represents whether this formula has been satisfied or not and, `Vals` holds the values assigned to variables so far in the processing of the verbal rule. Values are kept in `Vals` only when the formula in `Form` is satisfied, as indicated by `Truth`.

The clauses of `apply` which handle the existential and universal quantifiers are given below.

- ```
  apply(exists(Var,Restr,Form),Assign,Truth,Vals):-
     eval(Restr,Assign,Rval),
     quantifier_exists(exists,0
             ,Var,Form,Rval,Assign,Truth,[],Vals1),!,
     ((Truth = 1, Vals = Vals1)
     ;
     (Vals = [])).
  ```

- ```
  apply(forall(Var,Restr,Form),Assign,Truth,Vals):-
     eval(Restr,Assign,Rval),
     quantifier_forall(Var,Form,Rval,Assign,Truth,[],Vals1),!,
     ((Truth = 1, Vals = Vals1)
     ;
     (Vals = [])).
  ```

Quantifiers are given a structure of the form `Quant(Var,Restr,Form)` where `Quant` is the particular quantifier, `Var` is the variable bound by the quantifier, `Restr` is a restriction on the values `Var` can take and `Form` is the formula which is in the scope of the quantifier. In analysing quantifiers, a predicate `eval` determines the values that the variable `Var` may take after which a call to `quantifier_X` (where `X` is the particular quantifier in question) evaluates `Form` for each possible assignment to `Var`. The values returned in `Vals1` will be those assignments to variables in `Form` in which `Form` is satisfied for a particular value assigned to `Var`. For the existential quantifier `Truth` will be 1 if there is at least one such assignment to `Var` which satisfies `Form` while for the universal quantifier `Truth` will be 1 if all values assigned to `Var` satisfy `Form`.

When initially calling `apply`, `Assign` is provided with the values of the subject and object denotation sets as well as the particular verbal predicate being analysed. The application of

```
[donkey, [d1, d2, d3, d4, d5, d6, d7, d8, d9]].
[farmer, [f1, f2, f3, f4, f5, f6, f7, f8, f9]].
[own, [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]],
[[f4], [d4]], [[f5], [d5]], [[f6], [d6]],
[[f7], [d7]], [[f8], [d8]], [[f9], [d9]],
[[l1], [d1]], [[l2], [d2]], [[l3], [d3]]]].
[beat, [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f3], [d3]],
[[f4], [d4]], [[f5], [d5]], [[f6], [d6]],
[[f7], [d7]], [[f8], [d8]], [[f9], [d9]],
[[l1], [d1]], [[l2], [d2]], [[l3], [d3]]]].
```

Figure 7.5: An example model.

`apply` to the verification of verbal predicates for particular variable assignments returns in `Vals` the satisfying pair of sets or the empty set. From the values returned overall in `Vals` the sets from each denotation set along with the relationships between them can be collected together and a new graph including a possible new relational edge derived.

7.2.3 Worked Example

I will illustrate the implementation by providing a worked example within this section. Further, worked examples can be found in appendix C. I shall look at the following discourse.

(309) Every farmer owns a donkey. They beat them.

The model against which the interpretation will be given is shown in figure 7.5. For the purposes of this first discourse, the important fact is that every farmer owns exactly one donkey except farmer `f3` who owns two donkeys. Furthermore, every farmer beats a single donkey he owns, with farmer `f3` beating only one of the two donkeys he owns. The grammar used is similar to the one shown in appendix B. For the analysis of the first sentence the trace given below was derived. The system prompts the user to enter a system command or a sentence and also to use the command `reset` to begin a new discourse. The command `reset` is given as we wish to begin a new discourse. Next, the first sentence from (309) is given to the system. The system responds by stating that a successful parse has been found for this sentence and it displays the derived feature structure.

```
Enter a command or a sentence to analyse or just <RETURN> to finish
Type: reset   to start a new discourse
? reset
Enter a command or a sentence to analyse or just <RETURN> to finish
Type: reset   to start a new discourse
? every farmer owns a donkey
Successful Parse.
The derived feature set is shown below.
cat:s
```

```
head:
  syn:
    form:finite
    number:singular
  sem:
    control:
      subject:[(pred : every), (uniq : no), (pol : positive), (word : every),
        (reading : distributive), (number : singular)],pred:own,predicate:[(
        pol : positive), (scope : subjectwide)],object:[(pred : a), (uniq
        : no), (reading : distributive), (pol : positive), (word : a),
        (number : singular)],
    type:tv
    arg1:
      control:
        pred:every,uniq:no,pol:positive,word:every,reading:distributive,
        number:singular,
      type:det
      arg1:
        control:
          pred:farmer,word:farmer,number:singular,reading:distributive,pol:
        positive,uniq:no,
        type:n
    arg2:
      control:
        pred:a,uniq:no,reading:distributive,pol:positive,word:a,number:
        singular,
      type:det
      arg1:
        control:
          pred:donkey,word:donkey,number:singular,
        type:n
```

Next, the system states that it is providing a subject and object positive polarity distributive reading with no uniqueness restriction to the verbal predicate `own`. The validity of this reading can be checked by looking at the derived control information in the displayed feature structure. Finally, the derived top-level graph for this feature structure is displayed. The graph number is given, after which, the vertices, relational edges and anaphoric edges are displayed. The predicates from which these structures are derived are given in curved brackets. In this case, there are two vertices, one relational edge and no anaphoric edges.

```
---------------------------------------------
For the verbal predicate **own** the rule derived is:
Subject distributive, no uniqueness restriction.
Object distributive, no uniqueness restriction.
Positive Polarity
Subject wide scope
---------------------------------------------
Graph Derived
Graph 5
Vertices -
 Vertex 2 (donkey) containing [[d1], [d2], [d3], [d4], [d5], [d6], [d7],
```

```
        [d8], [d9]]
 Vertex 1 (farmer) containing [[f1, f2, f3, f4, f5, f6, f7, f8, f9]]
Relational Edges -
 Edge from 1 to 2 (own) : [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f3],
        [d3]], [[f4], [d4]], [[f5], [d5]], [[f6], [d6]], [[f7], [d7]],
        [[f8], [d8]], [[f9], [d9]]]
No anaphoric edges
```

Next, we enter the second sentence from (309), prompting the following trace. As before, a successful parse is found and the derived feature structure shown. Next, the system states that it is handling the anaphor *they*. At this point the system allows the user to enter the antecedents for this anaphor. This is accomplished in two stages. Firstly, graphs can be displayed to show the available vertices and graphs within the discourse and second some of these can be chosen as antecedents for the anaphor in question.

```
Enter a command or a sentence to analyse or just <RETURN> to finish
Type: reset   to start a new discourse
? they beat them
Successful Parse.
The derived feature set is shown below.
cat:s
head:
  syn:
    form:finite
    number:plural
  sem:
    control:
      subject:[(pred : they), (word : they)],pred:beat,predicate:[(pol
        : positive), (scope : subjectwide), (aarel : strong)],object:[(pred
        : them), (word : them)],
    type:tv
    arg1:
      control:
        pred:they,word:they,
      type:pro
    arg2:
      control:
        pred:them,word:them,
      type:pro
Handling the anaphor: they

Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
?
```

Choosing, the first option and looking at graph 5, provides the following trace. Graph number 5 is displayed. After which the user returns to the *Antecedent Choice Mode* main menu by typing

```
end.
```

```
? 1
Graph 5
Graph 4
Graph 3
Graph 2
Graph 1
Graph 0
Graph Display Mode
Possible commands:
  GNUM : display graph GNUM completely
  end  : quit graph display mode
? 5
Graph 5
Vertices -
 Vertex 2 (donkey) containing [[d1], [d2], [d3], [d4], [d5], [d6], [d7],
        [d8], [d9]]
 Vertex 1 (farmer) containing [[f1, f2, f3, f4, f5, f6, f7, f8, f9]]
Relational Edges -
 Edge from 1 to 2 (own) : [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f3],
        [d3]], [[f4], [d4]], [[f5], [d5]], [[f6], [d6]], [[f7], [d7]],
        [[f8], [d8]], [[f9], [d9]]]
No anaphoric edges
? end
Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
?
```

Next, an antecedent is chosen as shown below. Notice, that the antecedent is identified by giving a vertex and graph number pair. Certain useful functions are provide in order to manipulate antecedents to derive appropriate denotation sets for the anaphor. In this case, no such function is required and the denotation set for the anaphor is a copy of the denotation set for the antecedent. Essentially, the implementation allows the user to implement the $\mathcal{CONS}$ and $\mathcal{RES}$ functions from the framework.

```
Choose antecedents by giving vertex-graph pairs of the form [V,G],
where V is a vertex number and G is a graph number.
N-ary functions over several highlighted graphs can given, of the
form FUNC(A,B),
where FUNC is a function and A and B are other functions or
highlighted graphs.Available functions are:
  union : union
  sum   : summation
  ind   : individuation
PLACE A FULL STOP AT THE END OF THE EXPRESSION
 Examples: a) [2,5].
           b) sum([2,5],union([1,3],[2,3])).
|: [1,5].
```

A similar procedure is carried out for the anaphor *them* as illustrated below. Except in this case, we wish the antecedent to be vertex 2 from graph 6, graph 6 being the graph derived from the analysis of the anaphor *they*.

```
Handling the anaphor: them

Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
? 1
Graph 6
Graph 5
Graph 4
Graph 3
Graph 2
Graph 1
Graph 0
Graph Display Mode
Possible commands:
  GNUM : display graph GNUM completely
  end  : quit graph display mode
? 6
Graph 6
Vertices -
 Vertex 3 (they) containing [[f1, f2, f3, f4, f5, f6, f7, f8, f9]]
 Vertex 2 (donkey) containing [[d1], [d2], [d3], [d4], [d5], [d6], [d7],
        [d8], [d9]]
 Vertex 1 (farmer) containing [[f1, f2, f3, f4, f5, f6, f7, f8, f9]]
Relational Edges -
 Edge from 1 to 2 (own) : [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f3],
        [d3]], [[f4], [d4]], [[f5], [d5]], [[f6], [d6]], [[f7], [d7]],
        [[f8], [d8]], [[f9], [d9]]]
Anaphoric Edges -
 Edge from 3 to 1


? end
Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
? 2
Choose antecedents by giving vertex-graph pairs of the form [V,G],
where V is a vertex number and G is a graph number.
N-ary functions over several highlighted graphs can given, of the
form FUNC(A,B),
where FUNC is a function and A and B are other functions or
highlighted graphs.Available functions are:
  union : union
  sum   : summation
```

```
  ind   : individuation
PLACE A FULL STOP AT THE END OF THE EXPRESSION
 Examples: a) [2,5].
           b) sum([2,5],union([1,3],[2,3])).
|: [2,6].
```

After this, the system carried out the verbal analysis of the *beat* relation, which for the purposes of this run has been forced to give a *weak* anaphor-antecedent relation. The final graph is then displayed, as shown below.

```
---------------------------------------------
For the verbal predicate **beat** the rule derived is:
Subject distributive, no uniqueness restriction.
Object distributive, no uniqueness restriction.
Positive Polarity
Subject wide scope
---------------------------------------------
Graph Derived
Graph 8
Vertices -
 Vertex 4 (them) containing [[d1], [d2], [d4], [d5], [d6], [d7], [d8],
        [d9]]
 Vertex 3 (they) containing [[f1, f2, f3, f4, f5, f6, f7, f8, f9]]
 Vertex 2 (donkey) containing [[d1], [d2], [d4], [d5], [d6], [d7], [d8],
        [d9]]
 Vertex 1 (farmer) containing [[f1, f2, f3, f4, f5, f6, f7, f8, f9]]
Relational Edges -
 Edge from 3 to 4 (beat) : [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f4],
        [d4]], [[f5], [d5]], [[f6], [d6]], [[f7], [d7]], [[f8], [d8]],
        [[f9], [d9]]]
 Edge from 1 to 2 (own) : [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f3],
        [d3]], [[f4], [d4]], [[f5], [d5]], [[f6], [d6]], [[f7], [d7]],
        [[f8], [d8]], [[f9], [d9]]]
Anaphoric Edges -
 Edge from 3 to 1
 Edge from 4 to 2
```

If we reenter the second sentence of (309) but this time under a situation in which the verbal predicate `beat` is given a strong anaphor antecedent relation the following graph is derived.

```
---------------------------------------------
For the verbal predicate **beat** the rule derived is:
Subject distributive, no uniqueness restriction.
Object distributive, no uniqueness restriction.
Positive Polarity
Subject wide scope
---------------------------------------------
Strong Anaphor-Antecedent Relation applied.
Graph Derived
Graph 11
```

```
Vertices -
 Vertex 6 (them) containing []
 Vertex 5 (they) containing []
 Vertex 2 (donkey) containing []
 Vertex 1 (farmer) containing []
No relational edges
No anaphoric edges
```

As farmer `f3` does not beat both his donkeys the graph derived is empty, describing a false interpretation.

Chapter 8

Conclusion

This thesis has developed a novel model-theoretic semantic framework of discourse anaphora, Graph-Theoretic Semantics. The framework does not intend to prescribe a particular anaphoric theory but instead should be treated as a framework for semantic anaphoric analysis through which particular anaphoric theories can be implemented. That is, the thesis has proposed that semantic theories of discourse anaphora can profit by separating the theoretical framework for the semantic analysis of discourse and anaphora from the implementation of particular constraint theories of anaphoric reference. The framework provides the possibility for locating anaphoric constraints in both the representational and denotational domains.

The framework utilizes a feature-based semantic representation language which provides a powerful means of representing constraints on the interpretation of predicates as well as being a flexible representational formalism for incorporating future interpretational requirements derived from extending the coverage of the basic framework.

The locus for anaphoric information in GTS contrasts with that of other prominent model-theoretic semantic theories of anaphora. In DRT anaphoric information is manipulated within the representational domain, while in DPL anaphoric information is manipulated in the boundary between the representational and denotational domains. In GTS, anaphoric information is situated and manipulated solely within the denotational domain via the construction of denotation graphs. The determination of truth with respect to an interpretation is predicated on the derived graph structures. Unlike other theories such as Montague semantics which are type-theoretically organised so that sentences denote truth values, GTS is untyped. All syntactic constituents when appropriately interpreted derive graph structures as denotations. Deciding whether a declarative sentence makes a truthful statement is determined by examining the graph-theoretic denotation derived for that sentence by the semantic interpretation. In this respect the framework's treatment of truth determination is similar to DPL where truth is determined by examining the assignment functions derived from the interpretation of the logical representation constructed for a declarative sentence.

The denotational structures utilized to hold anaphoric information are graph-based. These graphs when treated as constraint networks allow the proper treatment of anaphor-antecedent relations and the retention of only relevant information within graphs as they are incrementally derived from previous graphs in the semantic analysis.

The utilization of constraint networks allows the variety of algorithms developed in constraint satisfaction research, as discussed in section 7.1.2 of chapter 7, to be utilized for the semantic analysis of natural language. In this respect the thesis has helped expand Haddock's "borderland" between natural language semantics and constraint network research mentioned in the introduction. In more concrete terms the thesis has illustrated how the analysis of generalized quantifiers and plural anaphoric reference can be treated within a theoretical framework based around constraint networks.

Chapter 1 provided a set of broad methodological and computational concerns that were highlighted as being of importance. In this section, I will review GTS against each of these concerns.

- Compositionality.
  GTS is a compositional framework for the semantic analysis of discourse anaphora. I have shown that a compositional framework can be derived for anaphoric problems which have previously proved difficult to handle in a compositional manner. However, a central theme of the debate concerning compositionality was the related concern for the existence or not of non-eliminable representations. Of the two theories central to this debate, DRT is non-compositional and derives non-eliminable representations while DPL is compositional and seems not to derive non-eliminable representations. However, one could argue that DPL does derive non-eliminable assignment functions. GTS derives non-eliminable denotations in the form of denotation graphs. The consistent theme is that any semantics of discourse anaphora will require non-eliminable structures in which information derived from the interpretation of a discourse is kept.

- Availability of anaphoric information.
  GTS contrasts with other semantic anaphoric theories in the abundance of anaphoric information it makes potentially available for anaphoric reference. Every syntactic constituent in a discourse (including sub-sentential constituents) will derive a denotation graph which will be placed in the discourse space which describes the anaphoric information derived from a discourse. A particular anaphoric theory based on the GTS framework may wish to reject the majority of this information, but nevertheless it is potentially available for use. Furthermore, the feature-based semantic representation language offers a wide variety of possible interpretations for the semantic analysis of a discourse. For instance, transitive verbal predicates can be read distributively or collectively, can be given subject or object wide scope, have uniqueness restrictions imposed on them and be given a variety of negative readings, e.g., sentence, verb phrase or verb negation. All these possibilities are determined solely by the values given to features describing a transitive verbal predicate. Under a particular syntactic analysis, only a **single** semantic representation structure is derived. Different interpretations of this structure are solely given via values to features within the particular semantic representation. This contrasts with most other theories in which different interpretations of a single syntactic constituent are identified via global structural changes in the semantic representation for this constituent. Furthermore, each feature plays a clear role in the determination of the global semantic interpretation.

- Flexibility.
  GTS utilizes a feature based representation structure which allows the semantics to have a very flexible base in which changes can be made with least disturbance to the existing

structure. This allows the GTS framework to be easily modifiable to cope with new semantic interpretations. For example, the extension of the possible verbal readings to include a cumulative reading would require no more than the creation of a new feature value for the reading feature and the construction of an appropriate verbal interpretation rule[1]. GTS also clearly separates the construction of anaphoric information from the accessibility of that information. As far as anaphor-antecedent relations are concerned GTS provides the four basic bound readings (weak, strong, unique antecedent, unique anaphor) as well as a referential reading of an anaphor. The controversies over the correct reading of donkey sentences has illustrated that it is unwise for a semantic theory to prejudge the data and limit itself to only a subset of the available anaphoric relations. In an attempt not to fall into one of Kaplan's seductions GTS is noncommittal on the correct anaphoric readings for particular discourses, especially those involving donkey sentences, and therefore prefers to provide the flexibility to cover the different possibilities within a framework that can as well be extended to cover interpretational possibilities not discussed in the present thesis.

- Constraint mechanisms.
  In chapter 6, it was mentioned that GTS can be viewed in two ways. As a framework of anaphoric analysis and as a framework for implementing particular anaphoric theories. Following the line taken in the last section, GTS does not impose a particular anaphoric theory but provides the means of deriving particular anaphoric theories via the mechanisms of constraint made available within the framework. In chapter 6 in sections 6.2.3 and 6.2.4 example constraint theories were given. In section 6.2.3, Kanazawa's constraints for deriving the appropriate reading of donkey sentences was implemented within the feature-based semantic representation via a particular PATR grammar. In section 6.2.4, denotational constraints for semantic number restrictions were provided. The use of feature-based representations within unification grammars has provided effective mechanisms for syntactic description and constraint. GTS provides the possibility of utilizing these methods within the semantic domain.

- Computational usefulness.
  GTS is a semantic framework of anaphora which purports not only to satisfy the linguistic constraints of anaphora but also to be a computational framework. In this, latter respect, the framework is intended to be computationally useful. The possible meanings of a computationally useful semantic framework were stated in chapter 1 and are given again below.

  - Computationally tractable.
  - Ease of Implementation.
  - Ease of integration with existing theories.

  The previous chapter has look informally at the possible computational complexity of the framework. It was shown that solving the constraint networks derived from the denotation graphs was likely to be NP-complete and thus provide a severe limitation to the computational tractability of the framework unless linguistic evidence is found to show that only some restricted class of constraint satisfaction problems can be derived from the semantic

[1] A possible verbal interpretation rule for a cumulative reading was given in section 6.4.2 of chapter 6.

analysis of discourses. Although a negative result, it does present a challenge to other semantic theories of anaphora to determine whether the probable intractability is an innate quality of the particular problems of anaphora being studied or is simply an expression of the manner in which the task has been approached within GTS. However, GTS does allow through its extensive use of constraint graphs the utilization of techniques from constraint satisfaction research to solve these graphs in an efficient manner in certain situations.

The ease of any implementation of the framework is greatly helped by the compositional nature of the analysis. An example implementation was described in the last chapter.

The utilization of feature structures suggests that the framework can be readily integrated with the present day feature-based unification grammars. Indeed, the simple unification grammar formalism PATR has been used extensively to illustrate the mechanism of such an integration.

8.1 Critical Analysis and Extensions

I shall begin by making a few critical comments concerning the GTS framework.

- The GTS framework as outlined within this work covers only a small set of syntactic constructions. Other complex semantic problems such as coordination, tense, prepositional phrases, and gaps have not been discussed. However, I will look at the denotation structures required for simple forms of coordination in section 8.1.1. Unfortunately, as GTS is a novel semantic framework, its novelty is a hindrance with respect to determining whether it can be extended to cover the complexity of the areas mentioned above.

- The connection between anaphor and antecedent within the semantics is expressed purely at the denotational level. However, most contemporary theories, such as DRT and DPL, express the connection between an anaphor and its antecedent at the representational level. That is, in GTS, looking at the semantic representation for a sentence, or the set of semantic representations for a discourse provides no information as to which anaphors reference which antecedents. This information is only derived through an interpretation of the sentence or discourse. However, the function $\mathcal{RES}$ provides a model-independent notion of anaphor-antecedent connection by specifying in any situation a particular resolution for an anaphor.

- By utilizing a function $\mathcal{RES}$ to determine a particular set of antecedents for an anaphor I am assuming that a unique set of antecedents can always be determined. This is probably a simplification, and the function $\mathcal{RES}$ could be converted to a relation in which several sets of anaphor-antecedent pairs are returned in ambiguous circumstances. However, the framework then needs to be changed to handle each possibility. This may require the semantic interpretation function as a whole being converted to a relation rather than a function.

In the following section I will look at the denotation structures needed to cover simple coordinated sentences. I will then outline further how alternative structures could be given to the discourse space other than the simple set-theoretic one utilized in the previous chapters.

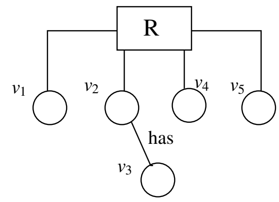


Figure 8.1: The denotation graph describing (310) with respect to a suitable model.

8.1.1 Denotational Structures for Coordination

I will look at how GTS might be extended to handle simple forms of coordination involving noun phrases and verb phrases. I will begin by illustrating the denotation structures that might be derived from the example given below.

(310) Every farmer or some peasant who has some money buys a donkey or steals a horse.

The denotation graph describing (310) in a model which satisfies the interpretation of this sentence is illustrated in figure 8.1. Coordinated sentences derive n-ary relational edges. Each nominal argument derives, as in the standard GTS, a separate vertex. Whereas before a transitive verbal predicate always had two nominal arguments, with coordination a transitive verbal predicate may have any number of arguments. Thus, instead of transitive verbal predicates deriving binary relational edges they derive n-ary relational edges connecting all their arguments. Furthermore, with the coordination of verb phrases a verbal predicate may not represent a single verbal relation but any number of verbal relations. As (310) does not concern a particular single verbal relation the resulting 4-ary edge has been given a generic name, *R*. This edge will specify the 4-ary (buying and stealing) relationships between the farmers (in v_1), the peasants (in v_2), the donkeys (in v_4) and the horses (in v_5). The basic interpretational line taken is to derive the arguments to a verbal predicate and then derive an n-ary relation describing the relationships between the coordinated arguments.

Anaphoric reference to these structures can follow in the usual manner. Shown again below is the sentence in (310) followed by two alternative continuation sentences.

(311) Every farmer or some peasant who has some money buys a donkey or steals a horse.

(312) They beat them.

(313) The peasants beat the horses.

In 312, I assume *they* refers to the farmers and peasants and *them* refers to the donkeys and horses. The determination of anaphoric circuits must take into account the fact that an n-ary constraint allows each of its n arguments to be connected to the other n-1 arguments. Given this, circuits can be determined as usual. For example, the graph in figure 8.2 has four possible anaphoric circuits, shown below.

- v_6, v_1, v_4, v_7, v_6
- v_6, v_2, v_4, v_7, v_6

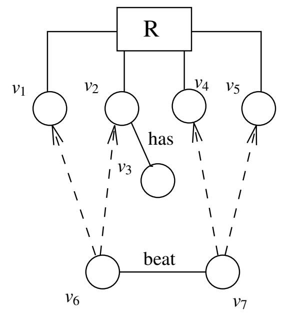


Figure 8.2: The denotation graph describing (311) followed by (312) with respect to a satisfying model.

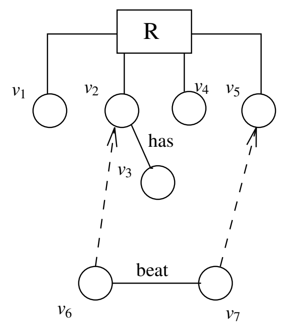


Figure 8.3: The denotation graphs describing (311) followed by (313) with respect to a satisfying model.

- v_6, v_1, v_5, v_7, v_6
- v_6, v_2, v_5, v_7, v_6

Weak and strong anaphor-antecedent relations can be provided, enforcing each farmer and peasant to beat at least one donkey or horse, or beat every donkey and horse, respectively. Continuing (311) with (313) derives in a satisfying model a denotation graph shown diagrammatically in figure 8.3. Again, both weak and strong anaphor-antecedent relations can be derived to enforce each peasant to beat at least one horse stolen, or beat every horse stolen[2].

Anaphora within and between conjuncts could also be handled, as illustrated by the following example.

(314) Every farmer buys a cart and a horse which pulls it.

In a suitable satisfying model, (314) will be interpreted to describe a denotation graph of the structure shown diagrammatically in figure 8.4. The vertex v_1 contains the farmers while the vertex v_2 contains the horses and v_3 the carts bought by the farmers. The vertex v_4 describes the carts anaphorically referred to by the anaphor *it*.

Another example will illustrate analysis of reflexives which refer to coordinated material, as shown below.

[2] I am assuming here that the definite determiner is being analysed in a similar way to a pronoun, probably in this case checking that all the referenced individuals are *peasants*.

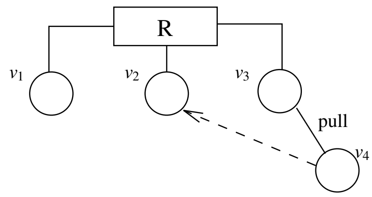


Figure 8.4: The denotation graph describing (314) with respect to a satisfying model.

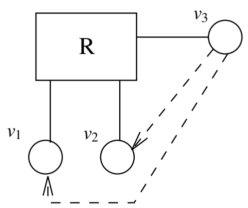


Figure 8.5: The denotation graph describing (315) with respect to a satisfying model.

(315) Every lawyer and some solicitors love themselves.

The denotation graph constructed for (315) is shown in figure 8.5, where v_3 are the lawyers and v_4 are the solicitors. There are two possible anaphoric circuits in this graph, v_1, v_3, v_1 and v_2, v_3, v_2. In finding a consistent labelling of this graph the first circuit will verify the anaphoric labelling constraint if v_1 and v_3 contain the same solicitor, the other circuit will verify a labelling in which v_1 and v_3 are the same lawyer.

This section has only sketched the possible manner in which GTS could handle coordinated linguistic constituents. Extensive further work is required to expand this into a viable account.

8.1.2 Alternative Structures for the Discourse Space

Throughout the discussion of the GTS framework a set-theoretic structure was utilized for the discourse space, in effect simply a set of denotation graphs. However, in section 6.2.1 of chapter 6 it was suggested that to allow for more complex structural constraints as employed by Kamp in DRT and Grosz with her use of discourse segmentation a more structurally complex discourse space would be required.

In order to accomplish this we would need to redefine the operators used by GTS to access and manipulate the discourse space. Within the framework as it stands, with the discourse space being represented as a set, these operators are the standard set-theoretic operators. In particular, the following operations appear within the semantic interpretation rules.

- $\mathcal{D} \cup \mathcal{D}'$, where $\cup$ is set-theoretic union.
- $G \in \mathcal{D}$, where $\in$ is set-theoretic membership

From this we can view the discourse space as a triple $\langle \mathcal{D}, \cup, \in \rangle$. However, an alternative utilization of GTS could redefine this triple in order to provided a more structurally complex discourse space, although the meta-level properties of membership and union would need to be retained.

In principle, the interpretation rules could be left descriptively unaltered. This would mean that the operations $\cup$ and $\in$ would for discourse spaces have alternative meanings to their traditional set-theoretic ones. The operations $\cup$ and $\in$ used elsewhere within the semantics would have their traditional set-theoretic meanings.

8.2 Final Comment

This thesis has presented a new semantic framework of discourse anaphora. The framework has addressed certain methodological, empirical and computational difficulties in the analysis of discourse anaphora. Methodologically, the framework is compositional and is designed to be intrinsically flexible while maintaining a clear separation between the different theoretical components of a semantic framework of anaphora. Empirically, the framework provides a flexible and extendible base allowing the provision of a wide range of readings to both the anaphoric and non-anaphoric components of a discourse. Computationally, the framework utilizes unification features structures for its representation, a type of representational structure prevalent in present day research. The framework utilizes graph-theoretic structures for its denotations. These structures are treated as describing constraint satisfaction problems during the interpretational process. The use of constraint networks allows results from constraint satisfaction research to be utilized profitably for computational linguistic purposes.

References

Ascher, N. 1986. Belief in Discourse Representation Theory. *Journal of Philosophy*, **15**, 127–189.

Barwise, J. 1987. Noun Phrases, Generalized Quantifiers, and Anaphora. *In:* Gardenfors, P. (ed), *Generalized Quantifiers*. Reidel, Dordrecht.

Barwise, J., & Cooper, R. 1981. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy*, **4**, 159–219.

Beaver, D. 1991. DMG through the Looking Glass. *Pages 135–153 of:* Stokhof, M., Groenendijk, J., & Beaver, D. (eds), *Quantification and Anaphora I*. DYANA Deliverable 2.2.A (Revised Version), Edinburgh University, Centre for Cognitive Science.

Bessiere, C. 1991. Arc-Consistency in Dynamic Constraint Satisfaction Problems. *Pages 221–226 of: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*.

Carter, D. 1987. *Interpreting Anaphors in Natural Language Text*. Ellis Horwood, Chichester.

Chierchia, G. 1991. Anaphora and Dynamic Logic. *Pages 37–79 of:* Stokhof, M., Groenendijk, J., & Beaver, D. (eds), *Quantification and Anaphora I*. DYANA Deliverable R2.2.A, Edinburgh University, Centre for Cognitive Science.

Chierchia, G. 1992a. Anaphora and Dynamic Binding. *Linguistics and Philosophy*, **15**(2), 111–183.

Chierchia, G. 1992b. *Dynamic Property Theory*. Fourth European Summer School in Logic, Language and Information, Essex University, Colchester, UK. (Lecture notes).

Chomsky, N. 1981. *Lectures on Government and Binding*. Foris Publications.

Cook, S. 1971. The Complexity of Theorem-Proving Procedures. *Pages 151–158 of: Proceedings of the Third Annual ACM Symposium on the Theory of Computing*.

Cooper, M. 1989. An Optimal k-Consistency Algorithm. *Artificial Intelligence*, **41**, 89–95.

Cooper, M., Cohen, D., & Jeavons, P. 1994. Characterising Tractable Constraints. *Artificial Intelligence*, **65**, 347–361.

Cooper, R. 1979. The Interpretation of Pronouns. *Pages 61–92 of:* Heny, F., & Schnelle, H. (eds), *Syntax and Semantics 10: Selections from the Third Groningen Round Table*. Academic Press, New York.

Cooper, R. 1991. Donkeys and Situations. *Pages 155–175 of:* Stokhof, M., Groenendijk, J., & Beaver, D. (eds), *Quantification and Anaphora I*. DYANA Deliverable R2.2.A, Edinburgh University, Centre for Cognitive Scince.

Cox, C. 1995. Constraint Networks for Natural Language Semantics. *In: Proceedings of CNIASE 95*. Forthcomming.

Dalrymple, M. 1993. *The Syntax of Anaphoric Binding*. CSLI Lecture Notes, Number 36.

Dechter, R., & Pearl, J. 1988. Network-Based Heuristics for Constraint-Satisfaction Problems. *Artificial Intelligence*, **34**, 1–38.

Dekker, P. 1991. The Scope of Negation in Discourse. *Pages 79–134 of:* Stokhof, M., Groenendijk, J., & Beaver, D. (eds), *Quantification and Anaphora I*. DYANA Deliverable 2.2.A (Revised Version), Edinburgh University, Centre for Cognitive Science.

Dowty, D., Wall, R., & Peters, S. 1981. *Introduction to Montague Semantics*. D. Reidel Publishing Company.

Elworthy, D. 1993. *The Semantics of Noun Phrase Anaphora*. Ph.D. thesis, University of Cambridge, Computer Laboratory. Published as Computer Laboratory Technical Report No. 289.

Evans, G. 1977. Pronouns, Quantifiers and Relative Clauses. *Canadian Journal of Philosophy*, **7**, 467–797.

Evans, G. 1980. Pronouns. *Linguistic Inquiry*, **11**, 337–362.

Evans, G. 1985. *Collected works*. Clarendon Press, Oxford.

Fenstad, J., Halvorsen, T., Langholm, T., & van Benthem, J. 1987. *Situations, Language and Logic*. Dordrecht: Reidel.

Frege, G. 1892. Uber Sinn und Bedeutung. *Zeitschrift fur Philosophie und Philosophische Kritik*, **100**, 25–50.

Frege, G. 1952. On Sense and Reference. *Pages 56–78 of:* Geach, P., & Black, M. (eds), *Translations from the Philosophical Writings of Gottlob Frege*. Oxford, Basil Blackwell.

Freuder, E. 1978. Synthesizing Constraint Expressions. *Communications of the ACM*, **21**, 958–966.

Freuder, E. 1982. A Sufficient Condition of Backtrack-Free Search. *Journal of the ACM*, **29**(1), 24–32.

Garey, M., & Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.

Gazdar, G., Klein, E., Pullum, G., & Sag, I. 1985. *Generalized Phrase Structure Grammar*. Blackwell, Oxford.

Geach, P. 1962. *Reference and Generality*. Cornell University Press, Ithaca.

Gibbons, A. 1985. *Algorithmic Graph Theory*. Cambridge University Press.

Groenendijk, J., & Stokhof, M. 1990a. Dynamic Montague Grammar. *In:* Kalman, L., & Polos, L. (eds), *Papers from the Second Symposium on Logic and Language*. Akademiai Kiado, Budapest.

Groenendijk, J., & Stokhof, M. 1990b. Dynamic Predicate Logic. *Pages 53–106 of:* van Bentham, J. (ed), *Partial and Dynamic Semantics I*. DYANA Deliverable R2.1.A, Edinburgh University, Centre for Cognitive Science.

Groenendijk, J., & Stokhof, M. 1991a. Dynamic Montague Grammar. *Pages 1–37 of:* Stokhof, M., Groenendijk, J., & Beaver, D. (eds), *Quantification and Anaphora I*. DYANA Deliverable 2.2.A (Revised Version), Edinburgh University, Centre for Cognitive Science.

Groenendijk, J., & Stokhof, M. 1991b. Dynamic Predicate Logic. *Linguistics and Philosophy*, **14**, 39–100.

Grosz, B., & Sidner, C. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, **12**(3), 175–204.

Haddock, N. 1988. *Incremental Semantics and Interactive Syntactic Processing*. Ph.D. thesis, University of Edinburgh.

Haddock, N. 1992. Semantic Evaluation as Constraint Network Consistency. *Pages 415–420 of: Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*.

Han, C., & Lee, C. 1988. Comments on Mohr and Henderson's Path Consistency Algorithm. *Artificial Intelligence*, **36**, 125–130.

Haralick, R., Davis, L., & Rosenfeld, A. 1978. Reduction Operations for Constraint Satisfaction. *Information Sciences*, **14**, 199–219.

Heim, I. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts, Amherst.

Heim, I. 1988. *The Semantics of Definite and Indefinite Noun Phrases*. Garland, New York.

Heim, I. 1990. E-Type Pronouns and Donkey Anaphora. *Linguistics and Philosophy*, **13**(2), 137–178.

Hentenryck, P., Deville, Y., & Teng, C. 1992. A Generic Arc-Consistency Algorithm and its Specializations. *Artificial Intelligence*, **57**, 291–321.

Higginbotham, J., & May, R. 1981. Questions, Quantifiers and Crossing. *The Linguistic Review*, **1**, 41–79.

Hirst, G. 1981. *Anaphora in Natural Language Understanding*. Berlin, Springer-Verlag.

Hobbs, J. 1986. Resolving Pronoun References. *Pages 339–352 of:* B. Grosz, K. S. J. (ed), *Readings in Natural Language Processing*. Morgan Kauffman, Los Altos, California.

Janssen, T. 1986. *Foundations and Applications of Montague Grammar*. CWI, Amsterdam.

Kadmon, N. 1990. Uniqueness. *Linguistics and Philosophy*, **13**, 273–324.

Kamp, H. 1981. A Theory of Truth and Semantic Representation. *Pages 277–322 of:* Groenendijk, J., Janssen, T., & Stokhof, M. (eds), *Formal Methods in the Study of Language*. Amsterdam Mathematical Centre Tracts Vol 135.

Kamp, H. 1985. Context, Thought and Communication. *Pages 239–261 of: Proceedings of the Aristotelian Society, New Series, Vol. LXXXV*.

Kamp, H. 1990. Comments on Groenendijk and Stokhof: Dynamic Predicate Logic. *Pages 109–131 of:* van Bentham, J. (ed), *Quantification and Anaphora II*. DYANA Deliverable R2.1.A, Edinburgh University, Centre for Cognitive Science.

Kamp, H. 1991. Uniqueness Presuppositions and Plural Anaphora in DTT and DRT. *Pages 177–189 of:* Stokhof, M., Groenendijk, J., & Beaver, D. (eds), *Quantification and Anaphora I*. DYANA Deliverable R2.2.A, University of Edinburgh, Centre for Cognitive Science.

Kamp, H., & Reyle, U. 1990. *From Discourse to Logic*. Second European Summer School in Logic, Language and Information, Katholieke Univeriteit Leuven. (Lecture notes).

Kamp, H., & Reyle, U. 1993. *From Discourse to Logic*. Kluwer Academic Publishers.

Kanazawa, M. 1994. Weak vs Strong Readings of Donkey Sentences and Monotonicity Inference in a Dynamic Setting. *Linguistics and Philosophy*, **17**, 109–158.

Kaplan, R. 1987. *Linguistic Theory and Computer Applications*. Academic Press.

Kaplan, R., & Bresnan, J. 1982. Lexical Functional Grammar. *Pages 173–281 of:* Bresnan, J. (ed), *The Mental Representation of Grammatical Relations*. MIT Press.

Kay, M. 1985. Parsing in Functional Unification Grammar. *Pages 251–278 of:* Dowty, D., Karttunen, L., & Zwicky, A. (eds), *Natural Language Parsing*. Cambridge University Press.

Keenan, E., & Stavi, J. 1986. A Semantic Characterization of Natural Language Determiners. *Linguistics and Philosophy*, **9**, 253–326.

Kratzer, A. 1979. Conditional Necessity and Possibility. *Pages 117–147 of:* Bauerle, R., Egli, U., & von Stechow, A. (eds), *Semantics from Different Points of View*. Springer-Verlag, New York.

Landman, F. 1989a. Groups I. *Linguistics and Philosophy*, **12**, 559–605.

Landman, F. 1989b. Groups II. *Linguistics and Philosophy*, **12**, 723–744.

Lasnik, H. 1976. Remarks on Coreference. *Linguistic Analysis*, **2**, 1–22.

Link, G. 1983. The Logical Analysis of Plural and Mass Terms: a Lattice-theoretical Approach. *Pages 245–257 of:* Bauerle, R., Schwarze, C., & von Stechow, A. (eds), *Meaning, Use and Interpretation of Language*. Walter de Gruyter, Berlin.

Link, G. 1984. Hydras: On the Logic of Relative Clause Constructions with Multiple Heads. *Pages 245–257 of:* Landman, F., & Veltman, F. (eds), *GRASS Series. Vol 3: Varieties of Formal Semantics*. Foris, Dordrecht.

Link, G. 1987. Generalized Quantifiers and Plurals. *Pages 151–180 of:* Gardenfors, P. (ed), *Generalized Quantifiers*. D. Reidel Publishing Company.

Lonning, J. 1991. Among Readings. *Pages 37–52 of:* van der Does, J. (ed), *Quantification and Anaphora II*. DYANA Deliverable 2.2.B, Edinburgh University, Centre for Cognitive Science.

Mackworth, A. 1977. Consistency in Networks of Relations. *Artificial Intelligence*, **8**(1), 99–118.

Mackworth, A., & Freuder, E. 1985. The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems. *Artificial Intelligence*, **25**, 65–73.

Matilal, B., & Sen, P. 1990. The context principle and some Indian controversies over meaning. *Mind*.

Mohr, R., & Henderson, T. 1986. Arc and Path Consistency Revisited. *Artificial Intelligence*, **28**(2), 225–233.

Moltmann, F. 1992. Reciprocals and Same/Different: Towards a Semantic Analysis. *Linguistics and Philosophy*, **15**, 411–462.

Montague, R. 1974a. English as a Formal Language. *Pages 188–221 of:* Thomason, R. (ed), *Formal Philosophy: Selected Papers of Richard Montague*. New haven: Yale University Press.

Montague, R. 1974b. The Proper Treatment of Quantification in Ordinary English. *Pages 247–270 of:* Thomason, R. (ed), *Formal Philosophy: Selected Papers of Richard Montague*. New haven: Yale University Press.

Montanari, U. 1974. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Information Science*, **7**, 95–132.

Montanari, U., & Rossi, F. 1991. Constraint Relaxation may be Perfect. *Artificial Intelligence*, **48**, 143–170.

Neale, S. 1990. *Descriptions*. The MIT Press.

Nerbonne, J. 1992 (August). *A Feature-Based Syntax/Semantics Interface*. Tech. rept. RR-92-42. DFKI, DFKI, -Bibliothek-,PF 2080, D-6750 Kaiserslautern,Germany.

Pollard, C., & Sag, I. 1987. *Information-Based Syntax and Semantics, Vol 1.* CSLI Lecture Notes No. 13.

Reinhart, T. 1976. *The Syntactic Domain of Anaphora.* Ph.D. thesis, MIT, Cambridge, Mass.

Reinhart, T. 1983. *Anaphora and Semantic Interpretation.* Croom Helm.

Richards, B. 1984. On Interpreting Pronouns. *Linguistics and Philosophy*, **7**, 287–324.

Roberts, C. 1987. *Modal Subordination, Anaphora and Distributivity.* Ph.D. thesis, University of Texas, Austin.

Roberts, C. 1989. Modal Subordination and Pronominal Anaphora in Discourse. *Linguistics and Philosophy*, **12**(6), 683–721.

Russell, B. 1905. On Denoting. *Mind*, **14**, 479–493.

Scha, R. 1981. Distributive, Collective and Cummulative Quantification. *Pages 483–512 of:* Groenendijk, J., Janssen, T., & Stokhof, M. (eds), *Formal Methods in the Study of Language.* Amsterdam Mathematical Centre Tracts Vol 136.

Scha, R. 1991. Afterthoughts on Collections. *Pages 53–58 of:* van der Does, J. (ed), *Quantification and Anaphora II.* DYANA Deliverable 2.2.B, Edinburgh University, Centre for Cognitive Science.

Schubert, L., & Pelletier, F. 1989. Generically Speaking, or, Using Discourse Representation Theory to Interpret Generics. *In:* Chierchia, G., Partee, B., & Turner, R. (eds), *Properties, Types and Meaning, II.* Kluwer Academic Publishers, Dordrect.

Schwarzchild, R. 1992. Types of Plural Individuals. *Linguistics and Philosophy*, **15**, 641–675.

Sells, P. 1986. Coreference and Bound Anaphora: A Restatement of the Facts. *Pages 434–446 of:* Berman (ed), *NELS 16, University of Massachusetts, Amherst.*

Shieber, S. 1986. *An Introduction to Unification-Based Approaches to Grammar.* CSLI Lecture Notes Number 4.

Shieber, S., Uszkoreit, H., Pereira, F., Robinson, J., & Tyson, M. 1983 (November). *The PATR-II Formalism.* Tech. rept. Research on Interactive Acquisition and Use of Knowledge (SRI Project 1894). SRI International.

Sidner, C. 1983. Focusing in the Comprehension of Definite Anaphora. *Pages 267–330 of:* Brady, M., & Berwick, R. (eds), *Computational Models of Discourse.* MIT Press.

van den Berg, M. 1990. A Dynamic Predicate Logic for Plurals. *Pages 29–51 of:* Stokhof, M., & Torenvliet, L. (eds), *Proceedings of the Seventh Amsterdam Colloquium.*

van den Berg, M. 1993. A Direct Definition of Generalized Dynamic Quantifiers. *Pages 121–140 of:* Stokhof, M., & Torenvliet, L. (eds), *Proceedings of the Ninth Amsterdam Colloquium.*

van der Does, J. 1991. Among Collections. *Pages 1–36 of:* van der Does, J. (ed), *Quantification and Anaphora II*. DYANA Deliverable 2.2.B, Edinburgh University, Centre for Cognitive Science.

Webber, B. 1979. *A Formal Approach to Discourse Anaphora*. Garland Publishing Inc., New York.

Zadrozny, W. 1994. From compositional to systematic semantics. *Linguistics and Philosophy*, **17**, 329–342.

Zeevat, H. 1986. A Treatment of Belief Sentences in Discourse Representation Theory. *Pages 189–215 of:* Groenendijk, J., de Jongh, D., & Stokhof, M. (eds), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*. Foris, Dordrecht.

Zeevat, H. 1989. A Compositional Approach to Discourse Representation Theory. *Linguistics and Philosophy*, **12**, 95–131.

Zeevat, H. 1990. Static Semantics. *Pages 134–140 of:* van Bentham, J. (ed), *Quantification and Anaphora II*. DYANA Deliverable R2.1.A, Edinburgh University, Centre for Cognitive Science.

Appendix A

GTS Semantic Interpretation Rules

The complete set of semantic interpretation rules for the GTS semantic framework are given below.

The interpretation for a lexical noun predicate is given below, where M is a model, I is a set of identifiers, $\mathcal{CONS}$ is an anaphoric constraint function, $\mathcal{RES}$ is an anaphoric resolution function and α is a feature-based semantic representation.

$$\text{If} \begin{bmatrix} \text{control} & \begin{bmatrix} \text{pred} & \text{A} \\ \text{number} & \text{B} \end{bmatrix} \end{bmatrix}_n \sqsubseteq \alpha \text{ and } \begin{pmatrix} \text{PRED} = \alpha/\langle\text{control pred}\rangle \\ \text{NUM} = \alpha/\langle\text{control number}\rangle \end{pmatrix} \text{ then}$$
$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G[v],\mathcal{D} \cup \{G[v]\})\rangle$, where,

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$.
- $v = \langle i, \mathcal{C}\rangle$ where $\mathcal{C} = \begin{cases} \{X \subseteq F(\text{PRED}) \mid |X| = 1\} & \text{If NUM} = \text{singular} \\ \{X \subseteq F(\text{PRED}) \mid |X| \geq 1\} & \text{If NUM} = \text{plural} \end{cases}$

The interpretation of a determiner predicate is given below, where M is a model, I is a set of identifiers $\mathcal{CONS}$ is an anaphoric constraint function, $\mathcal{RES}$ is an anaphoric resolution function and α is a feature-based semantic representation.

$$\text{If} \begin{bmatrix} \text{control} & \begin{bmatrix} \text{pred} & \text{A} \\ \text{number} & \text{B} \\ \text{uniq} & \text{C} \\ \text{pol} & \text{D} \end{bmatrix} \\ \text{arg1} & \text{G} \end{bmatrix}_{det} \sqsubseteq \alpha \text{ and } \begin{pmatrix} \text{PRED} = \alpha/\langle\text{control pred}\rangle \\ \text{NUM} = \alpha/\langle\text{control number}\rangle \\ \text{ARG} = \alpha/\langle\text{arg1}\rangle \end{pmatrix} \text{ then}$$
$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G'',\mathcal{D}' \cup \{G''\})\rangle$ where

- $[\![\text{ARG}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$
- $\langle i, \mathcal{C}\rangle \in G'$, $S = \bigcup_{X \in \mathcal{C}} X$
  The vertex holding the information of the argument to the determiner is $\langle i, \mathcal{C}\rangle$. We take the union of the sets in $\mathcal{C}$.

- $$C' = \begin{cases} \{X \subseteq S | X = S\} & \text{If PRED} = \text{every} \\ \{X \subseteq S | |X| \geq \frac{1}{2}|S|\} & \text{If PRED} = \text{most} \\ \{X \subseteq S | |X| = 1\} & \text{If PRED} = \text{a} \\ \{X \subseteq S | |X| = 1\} & \text{If PRED} = \text{some}, \text{NUM} = \text{singular} \\ \{X \subseteq S | |X| > 1\} & \text{If PRED} = \text{some}, \text{NUM} = \text{plural} \\ \{S - X | X = \{\}\} & \text{If PRED} = \text{no} \\ \{S - X | X \subseteq S \wedge |X| < \frac{1}{2}|S|\} & \text{If PRED} = \text{few} \\ \{X \subseteq S | |X| = 2\} & \text{If PRED} = \text{two} \end{cases}$$
  From the set S we can determine the denotation set $\mathcal{C}'$ containing the witness sets for the particular quantifier in question.

- $G'' = G'[\langle i, \mathcal{C}' \rangle / \langle i, \mathcal{C} \rangle]$
  The graph G'' is the graph G' with the vertex $\langle i, \mathcal{C} \rangle$ replaced by the vertex $\langle i, \mathcal{C}' \rangle$.

The interpretation rule for proper names is given below where M is a model, I is a set of identifiers and α is a feature-based semantic representation.

$$\text{If} \begin{bmatrix} \text{control} & \begin{bmatrix} \text{pred} & \text{A} \\ \text{number} & \text{B} \\ \text{uniq} & \text{C} \\ \text{pol} & \text{D} \end{bmatrix} \end{bmatrix}_{pn} \sqsubseteq \alpha \text{ and PRED} = \alpha/\langle \text{control pred} \rangle \text{ then}$$

$[\![\alpha]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G[v], \mathcal{D} \cup \{G[v]\}) \rangle$, where

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$.
- $v = \langle i, \mathcal{C} \rangle$ where $\mathcal{C} = \{F(\text{PRED})\}$

The interpretation of referential and bound anaphors is given below, where M is a model, I is a set of identifiers, $\mathcal{CONS}$ is an anaphoric constraint function, $\mathcal{RES}$ is an anaphoric resolution function, $\mathcal{DIS}$ is a discourse context and α is a feature-based semantic representation from $\alpha_1, \alpha_2, ... \alpha_n$.

$$\text{If} \begin{bmatrix} \text{control} & \begin{bmatrix} \text{pred} & \text{PRO} \\ \text{anaphor} & \text{referential} \\ \text{number} & \text{NUM} \end{bmatrix} \end{bmatrix}_{pro} \sqsubseteq \alpha \text{ then}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G, \mathcal{D}), (i, G', \mathcal{D} \cup \{G'\}) \rangle$ where

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$
- $\mathcal{RES}(\mathcal{DIS}, \mathcal{CONS}(\alpha, \mathcal{D}, G)) = \langle \mathcal{C}, \mathcal{R} \rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function $\mathcal{RES}$ to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{CONS}$.

- $G' = G[\langle i, \mathcal{C}\rangle]$
  Create the graph for the anaphor by extending G with the anaphor vertex $\langle i, \mathcal{C}\rangle$.

$$\text{If} \left[\begin{array}{ll} \mathsf{control} & \left[\begin{array}{ll} \mathsf{pred} & \mathsf{PRO} \\ \mathsf{anaphor} & \mathsf{bound} \\ \mathsf{number} & \mathsf{NUM} \end{array} \right] \end{array} \right]_{pro} \sqsubseteq \alpha \text{ then}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G,\mathcal{D}),(i,G',\mathcal{D}\cup\{G'\})\rangle$ where

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$
- $\mathcal{RES}(\mathcal{DIS},\mathcal{CONS}(\alpha,\mathcal{D},G)) = \langle \mathcal{C},\mathcal{R}\rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function $\mathcal{RES}$ to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{CONS}$.
- $v = \langle i, \mathcal{C}\rangle$.
  The vertex for the anaphor is created.
- $A = \{\langle v, v_i\rangle | \langle v_i, G\rangle \in \mathcal{R}\}$ and $G_{t1} = \langle\{\},\{\},A\rangle$
  A is the set of anaphoric edges linking anaphor to antecedent, and a graph G_{t1} is created to hold these anaphoric edges.
- $G_{t2} = \bigcup_{G\in\mathcal{R}_G} G$
  A graph G_{t2} is created from the union of the antecedent graphs.
- $G' = G[v] \cup G_{t1} \cup G_{t2}$
  The graph for the anaphor is the union of the extension of the graph G with the anaphor vertex along with the graphs G_{t1} and G_{t2}.

The interpretation of a verbal predicate is given below, where M is a model and I is a set of identifiers, $\mathcal{CONS}$ is an anaphoric constraint function and $\mathcal{RES}$ is an anaphoric resolution function and α is a feature-based semantic representation.

$$\text{If} \left[\begin{array}{ll} \mathsf{control} & \left[\begin{array}{ll} \mathsf{subject} & \mathsf{S} \\ \mathsf{object} & \mathsf{O} \\ \mathsf{predicate} & \mathsf{P} \end{array} \right] \\ \mathsf{arg1} & \mathsf{X} \\ \mathsf{arg2} & \mathsf{Y} \end{array} \right]_{tv} \sqsubseteq \alpha, \text{ and} \left(\begin{array}{l} \mathsf{V} = \alpha/\langle \mathsf{control\ predicate\ pred}\rangle \\ \mathsf{ARG1} = \alpha/\langle \mathsf{arg1}\rangle \\ \mathsf{ARG2} = \alpha/\langle \mathsf{arg2}\rangle \\ \mathsf{CTRL} = \alpha/\langle \mathsf{control}\rangle \\ \mathsf{AAREL} = \alpha/\langle \mathsf{control\ aarel}\rangle \end{array} \right) \text{ then,}$$

$[\![\alpha]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1,\mathcal{D}_1),(i_1,G_4,\mathcal{D}_3\cup\{G_4\})\rangle$ where

- $[\![\mathsf{ARG1}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_1,\mathcal{D}_1),(i_1,G_2,\mathcal{D}_2)\rangle$

- $[\![\mathsf{ARG2}]\!]^{M,I,\mathcal{CONS},\mathcal{RES}} = \langle (G_2, \mathcal{D}_2), (i_2, G_3, \mathcal{D}_3) \rangle$

- $v = \langle i_1, \mathcal{C} \rangle$ where $\langle i_1, \mathcal{C} \rangle \in G_3$ and $v' = \langle i_2, \mathcal{C}' \rangle$ where $\langle i_2, \mathcal{C}' \rangle \in G_3$
  The vertices for each argument are determined via the identifiers i_1 and i_2.

- $R = \{\langle X, Y \rangle | \exists S_1 \in \mathcal{C}, \exists S_2 \in \mathcal{C}' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The relation R allows any pair of subsets from either argument.

- $R_a = \{\langle X, Y \rangle \in R | satis(G_3[\langle v, v', R \rangle], \mathcal{L}) \wedge \{\langle v, X \rangle, \langle v', Y \rangle\} \subseteq \mathcal{L}\}$
  The relation R_a limits the relation R by allowing only anaphorically acceptable pairs from R. This is determined via the relation satis (defined on page 86) over the graph G extended with an edge between the vertices v and v′ utilizing the relation R. The sets X and Y are labels for the vertices v and v′ respectively.

- If ϕ is the interpretation rule derived from CTRL then:
  - If SCOPE = subjectwide, $\phi' = (((\phi(\mathcal{C})), (\mathcal{C}')), (F(V)))$
  - If SCOPE = objectwide, $\phi' = (((\phi(\mathcal{C}')), (\mathcal{C})), (F(V)))$

  The verbal reading rule is derived, the arguments to ϕ being given in an order determined by the feature scope.

- There is a mapping from $\langle \mathcal{C}, \mathcal{C}', R_a \rangle$ to $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ where $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$ is the componentwise union of all triples, $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle, \mathcal{C}_t \subseteq \mathcal{C}, \mathcal{C}'_t \subseteq \mathcal{C}', R_t \subseteq R_a$ which minimally satisfy the interpretation rule ϕ'
  The set of triples $\langle \mathcal{C}_t, \mathcal{C}'_t, R_t \rangle$ which satisfy the verbal reading described by ϕ' are collected together in $\langle \mathcal{C}_s, \mathcal{C}_o, R' \rangle$.

- $v_s = \langle i_1, \mathcal{C}_s \rangle$ and $v_o = \langle i_2, \mathcal{C}_o \rangle$.
  New vertices are constructed.

- If $R' = \{\}$ then $G_4 = cons(G_3[v_s/v, v_o/v'])$ else
  $G_4 = cons(G_3[[v_s/v, v_o/v'][\langle v_s, v_o, R' \rangle])$
  If the derived relation R' is empty no relational edge is constructed between the new vertices. The function cons (defined on page 86) forces the new graphs to be maximally consistent.

Appendix B

PATR Grammar

```
RULE {sentence matrix}
 S -> NP VP:

   <S head> = <VP head>
   <S head syn form> = finite
   <VP subcat first> = <NP>
   <VP subcat rest> = end
   <S head sem control subject> = <NP head sem control>
   <NP head syn rel> = false.

RULE {sentence relative}
 S -> NP VP:
   <S head> = <VP head>
   <S head sem control subjrel> = <NP head sem control word>
   <S head syn form> = finite
   <S subcat> = <VP subcat>
   <NP head syn rel> = true
   <S head syn rel> = true.

Rule {transitive verb phrase}

VP_1 -> V NP:
   <VP_1 head> = <V head>
   <V subcat first> = <NP>
   <VP_1 subcat> = <V subcat rest>
   <VP_1 head sem control object> = <NP head sem control>.

Rule {Negative verb}

V_3 -> V_1 NEG V_2:
   <V_1 head form> = aux
   <V_2 head syn form> = base
   <V_3 head sem > = <V_2 head sem>
   <V_3 subcat> = <V_2 subcat>
   <V_3 subcat rest first head syn number> = <V_1 head syn number>
   <V_3 head sem control predicate pol> = negative.

Rule {Noun phrase}

NP -> Det Nbar:
 <NP head> = <Det head>
 <Det head syn number> = <Nbar head syn number>
 <NP head sem control number> = <Nbar head syn number>
 <NP head syn rel> = false
 <Det subcat first> = <Nbar>
```

```
  <Det subcat rest> = end.

Rule {Proper Noun}

NP -> PN:
 <NP head> = <PN head>
 <NP head syn rel> = false.

 Rule {Nbar lexical noun}

 Nbar -> N:
  <Nbar head> = <N head>.

 Rule {Relative clause combination}

 Nbar_1 -> N S:
   <Nbar_1 head> = <S head>
   <Nbar_1 head sem control subject> = <N head sem control>
   <S subcat first> = <N>
   <S head syn rel> = true
   <S subcat rest> = end.

Word who: <cat> = np
    <head syn rel> = true.

Word it: <cat> = np
         <head sem control pred> = it
         <head sem control word> = it
         <head sem type> = pro
         <head sem control number> = singular
         <head sem control anaphor> = bound
         <head syn number> = singular
         <head syn person> = third
         <head syn rel> = false.

Word they: <cat> = np
        <head syn number> = plural
        <head syn person> = third
        <head sem control pred> = they
        <head sem control word> = they
        <head sem type> = pro
        <head syn rel> = false.

Word them: <cat> = np
        <head syn number> = plural
        <head syn person> = third
        <head sem control pred> = them
        <head sem control word> = them
        <head sem type> = pro
        <head syn rel> = false.

Word himself: <cat> = np
        <head syn number> = singular
        <head syn person> = third
        <head sem control pred> = himself
        <head sem control word> = himself
        <head sem control anaphor> = bound
        <head sem type> = pro
        <head syn rel> = false.


Word does: <cat> = v
 <head syn form> = aux
```

```
 <head syn number> = singular.

Word do: <cat> = v
 <head syn form> = aux
 <head syn number> = plural.

Word not: <cat> = neg.

Word every: <cat> = det
        <head sem control pred> = every
        <head sem control uniq> = no
        <head sem control pol> = positive
        <head sem type> = det
        <head sem control word> = every
        <head sem control reading> = distributive
        <head sem arg1> = <subcat first head sem>
        <head syn number> = singular
        <subcat first cat> = nbar
        <subcat rest> = end.

Word most: <cat> = det
        <head sem control pred> = most
        <head sem control uniq> = no
        <head sem control pol> = positive
        <head sem type> = det
        <head sem control word> = most
        <head sem control reading> = distributive
        <head sem arg1> = <subcat first head sem>
        <head syn number> = plural
        <subcat first cat> = nbar
        <subcat rest> = end.

Word a: <cat> = det
        <head sem control pred> = a
        <head sem control uniq> = no
        <head sem control reading> = distributive
        <head sem control pol> = positive
        <head sem type> = det
        <head sem control word> = a
        <head sem arg1> = <subcat first head sem>
        <head syn number> = singular
        <subcat first cat> = nbar
        <subcat rest> = end.

Word farmer: <cat> = n
         <head sem control pred> = farmer
         <head sem control word> = farmer
         <head sem type> = n
         <head sem control number> = singular
         <head sem control reading> = distributive
         <head sem control pol> = positive
         <head sem control uniq> = no
         <head syn number> = singular
         <head syn person> = third.

Word farmers: <cat> = n
         <head sem control pred> = farmer
         <head sem control word> = farmer
         <head sem type> = n
         <head sem control number> = plural
         <head syn number> = plural
         <head syn person> = third.
```

```
Word donkey: <cat> = n
         <head sem control pred> = donkey
         <head sem control word> = donkey
         <head sem type> = n
         <head sem control number> = singular
         <head syn number> = singular.

Word owns: <cat> = v
        <head syn form> = finite
        <head syn number> = singular
        <head sem control pred> = own
        <head sem control predicate pol> = positive
        <head sem control predicate scope> = subjectwide
        <head sem type> = tv
        <head sem arg1> = <subcat rest first head sem>
        <head sem arg2> = <subcat first head sem>
        <subcat rest first head syn number> = singular
        <subcat rest first head syn person> = third
        <subcat rest rest> = end.

Word beats: <cat> = v
        <head syn form> = finite
        <head syn number> = singular
        <head sem control pred> = beat
        <head sem control predicate pol> = positive
        <head sem control predicate scope> = subjectwide
        <head sem control predicate aarel> = weak
        <head sem type> = tv
        <head sem arg1> = <subcat rest first head sem>
        <head sem arg2> = <subcat first head sem>
        <subcat rest first head syn number> = singular
        <subcat rest first head syn person> = third
        <subcat rest rest> = end.

Word beat: <cat> = v
        <head syn form> = finite
        <head syn number> = plural
        <head sem control pred> = beat
        <head sem control predicate pol> = positive
        <head sem control predicate scope> = subjectwide
        <head sem control predicate aarel> = strong
        <head sem type> = tv
        <head sem arg1> = <subcat rest first head sem>
        <head sem arg2> = <subcat first head sem>
        <subcat rest first head syn number> = plural
        <subcat rest first head syn person> = third
        <subcat rest rest> = end.


Word own: <cat> = v
        <head syn form> = base
        <head sem control pred> = own
        <head sem control predicate scope> = subjectwide
        <head sem type> = tv
        <head sem arg1> = <subcat rest first head sem>
        <head sem arg2> = <subcat first head sem>
        <subcat rest rest> = end.

end.
```

Appendix C

Further Worked Examples

In this appendix I will illustrate the implementation of GTS discussed in section 7.2 of chapter 7 with some further worked examples.

The first example I shall look at is a simple reflexive.

(316) Every farmer loves himself.

Assuming an interpretation relative to a model shown below.

- `[farmer, [f1, f2, f3]].`
  `[love, [ [[f1],[f1]], [[f2],[f2]], [[f3],[f3]] ]].`

An implementation run is shown below.

```
? every farmer loves himself
Successful Parse.
The derived feature set is shown below.
cat:s
head:
  syn:
    form:finite
    number:singular
  sem:
    control:
      subject:[(pred : every), (uniq : no), (pol : positive), (word : every),
        (reading : distributive), (number : singular)],pred:love,predicate:[(
        pol : positive), (scope : subjectwide)],object:[(pred : himself),
        (word : himself), (anaphor : bound)],
    type:tv
    arg1:
      control:
        pred:every,uniq:no,pol:positive,word:every,reading:distributive,
        number:singular,
      type:det
      arg1:
        control:
          pred:farmer,word:farmer,number:singular,reading:distributive,pol:
        positive,uniq:no,
        type:n
```

```
    arg2:
      control:
        pred:himself,word:himself,anaphor:bound,
      type:pro

Handling the anaphor: himself

Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
? 1
Graph 2
Graph 1
Graph 0
Graph Display Mode
Possible commands:
  GNUM : display graph GNUM completely
  end  : quit graph display mode
? 2
```

[1]

```
Graph 2
Vertices -
 Vertex 1 (farmer) containing [[f1, f2, f3]]
No relational edges
No anaphoric edges

? end
Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
? 2
Choose antecedents by giving vertex-graph pairs of the form [V,G],
where V is a vertex number and G is a graph number.
N-ary functions over several highlighted graphs can given, of the
form FUNC(A,B),
where FUNC is a function and A and B are other functions or
highlighted graphs.Available functions are:
  union : union
  sum   : summation
  ind   : individuation
  join  : joining sets from two denotation sets
PLACE A FULL-STOP AT THE END OF THE EXPRESSION
 Examples: a) [2,5].
           b) sum([2,5],union([1,3],[2,3])).
```

[2]

```
|: ind([1,2]).
---------------------------------------------
For the verbal predicate **love** the rule derived is:
Subject distributive, no uniqueness restriction.
```

```
Object distributive, no uniqueness restriction.
Positive Polarity
Subject wide scope
---------------------------------------------
```

[3]

```
Graph Derived
Graph 4
Vertices -
 Vertex 2 (himself) containing [[f1], [f2], [f3]]
 Vertex 1 (farmer) containing [[f1, f2, f3]]
Relational Edges -
 Edge from 1 to 2 (love) : [[[f1], [f1]], [[f2], [f2]], [[f3], [f3]]]
Anaphoric Edges -
 Edge from 2 to 1
```

At [1], the graph containing the antecedent is displayed. At [2], the denotation set for the anaphor is constructed from the individuals contained within vertex 1 of the graph displayed at [1]. At [3], the final graph derived from the analysis of this sentence is displayed. The graph's vertices are non-empty, identifying a truthful interpretation against the particular model given previously.

The next example is the extended donkey sentence, example (280) from section 6.4.2 of chapter 6.

(317) Every farmer who owns a donkey attacks a man who beats it.

The model against which this sentence will be interpreted is given below.

- ```
  [farmer, [f1, f2, f3]].
  [man, [m1, m2, m3]].
  [attack, [ [[f1],[m1]], [[f2],[m2]], [[f3],[m3]],
             [[f3],[m2]] ]].
  [own, [ [[f1], [d1]], [[f2], [d2]], [[f3], [d2]],
          [[f3], [d3]] ]].
  [beat, [[[m1], [d1]], [[m2], [d2]], [[m3], [d3]]]].
  ```

The implementation run is shown below.

```
? every farmer who owns a donkey attacks a man who beats it
Successful Parse.
The derived feature set is shown below.
cat:s
head:
  syn:
    form:finite
    number:singular
  sem:
    control:
      subject:[(pred : every), (uniq : no), (pol : positive), (word : every),
```
```

```
        (reading : distributive), (number : singular)],pred:attack,predicate:
        [(pol : positive), (scope : subjectwide), (aarel : weak)],object:[(
        pred : a), (uniq : no), (reading : distributive), (pol : positive),
        (word : a), (number : singular)],
    type:tv
    arg1:
      control:
        pred:every,uniq:no,pol:positive,word:every,reading:distributive,
        number:singular,
      type:det
      arg1:
        control:
          subject:[(pred : farmer), (word : farmer), (number : singular),
        (reading : distributive), (pol : positive), (uniq : no)],subjrel:[],
        pred:own,predicate:[(pol : positive), (scope : subjectwide)],object:[
        (pred : a), (uniq : no), (reading : distributive), (pol : positive),
        (word : a), (number : singular)],
        type:tv
        arg1:
          control:
            pred:farmer,word:farmer,number:singular,reading:distributive,pol:
        positive,uniq:no,
          type:n
        arg2:
          control:
            pred:a,uniq:no,reading:distributive,pol:positive,word:a,number:
        singular,
          type:det
          arg1:
            control:
              pred:donkey,word:donkey,number:singular,
            type:n
    arg2:
      control:
        pred:a,uniq:no,reading:distributive,pol:positive,word:a,number:
        singular,
      type:det
      arg1:
        control:
          subject:[(pred : man), (word : man), (number : singular), (reading
        : distributive), (pol : positive), (uniq : no)],subjrel:[],pred:beat,
        predicate:[(pol : positive), (scope : subjectwide)],object:[(pred
        : it), (word : it), (number : singular), (anaphor : bound)],
        type:tv
        arg1:
          control:
            pred:man,word:man,number:singular,reading:distributive,pol:
        positive,uniq:no,
          type:n
        arg2:
          control:
            pred:it,word:it,number:singular,anaphor:bound,
          type:pro
---------------------------------------------
For the verbal predicate **own** the rule derived is:
```

```
Subject distributive, no uniqueness restriction.
Object distributive, no uniqueness restriction.
Positive Polarity
Subject wide scope
---------------------------------------------

Handling the anaphor: it

Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
? 1
Graph 6
Graph 5
Graph 4
Graph 3
Graph 2
Graph 1
Graph 0
Graph Display Mode
Possible commands:
  GNUM : display graph GNUM completely
  end  : quit graph display mode
? 6
Graph 6
Vertices -
 Vertex 3 (man) containing [[m1], [m2], [m3]]
 Vertex 1 (farmer) containing [[f1, f2, f3]]
 Vertex 2 (donkey) containing [[d1], [d2], [d3]]
Relational Edges -
 Edge from 1 to 2 (own) : [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f3],
        [d3]]]
No anaphoric edges

? end
Antecedent Choice Mode
Possible commands:
 1 : Go to Graph Display Mode to show the present discourse space.
 2 : Choose some antecedents.
? 2
Choose antecedents by giving vertex-graph pairs of the form [V,G],
where V is a vertex number and G is a graph number.
N-ary functions over several highlighted graphs can given, of the
form FUNC(A,B),
where FUNC is a function and A and B are other functions or
highlighted graphs.Available functions are:
  union : union
  sum   : summation
  ind   : individuation
  join  : joining sets from two denotation sets
PLACE A FULL-STOP AT THE END OF THE EXPRESSION
 Examples: a) [2,5].
           b) sum([2,5],union([1,3],[2,3])).
|: [2,6].
```

```
---------------------------------------------
For the verbal predicate **beat** the rule derived is:
Subject distributive, no uniqueness restriction.
Object distributive, no uniqueness restriction.
Positive Polarity
Subject wide scope
---------------------------------------------
---------------------------------------------
For the verbal predicate **attack** the rule derived is:
Subject distributive, no uniqueness restriction.
Object distributive, no uniqueness restriction.
Positive Polarity
Subject wide scope
---------------------------------------------
Graph Derived
Graph 10
Vertices -
 Vertex 3 (man) containing [[m1], [m2], [m3]]
 Vertex 1 (farmer) containing [[f1, f2, f3]]
 Vertex 4 (it) containing [[d1], [d2], [d3]]
 Vertex 2 (donkey) containing [[d1], [d2], [d3]]
Relational Edges -
 Edge from 1 to 3 (attack) : [[[f1], [m1]], [[f2], [m2]], [[f3], [m2]],
        [[f3], [m3]]]
 Edge from 3 to 4 (beat) : [[[m1], [d1]], [[m2], [d2]], [[m3], [d3]]]
 Edge from 1 to 2 (own) : [[[f1], [d1]], [[f2], [d2]], [[f3], [d2]], [[f3],
        [d3]]]
Anaphoric Edges -
 Edge from 4 to 2
```