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

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 espe

Abstract

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 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

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, derivi

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

Anaphora and Semantics

This thesis is concerned with providing a formal semantic account of one aspect of discourse 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

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,) A e44422(c)5.64422(u)-4.on, ra404(h)-4.10691(h)-4.10914(e)-23534(u)-4.11026(

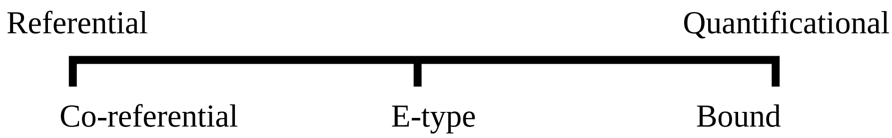


Figure 2.1: Anaphor-Antecedent Relations.

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].

2.2 Donkey Sentences

A great deal of theoretical discussion has centered around finding the proper analysis of one group

(16) ∀

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” 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).

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

x Donkeys Owned	*y Donkeys Beaten*	*Reading*
$x = 1$	$y = 1$	Unique Antecedent
$x \geq 1$	$y = 1$	Unique Anaphor

.

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.

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 fr

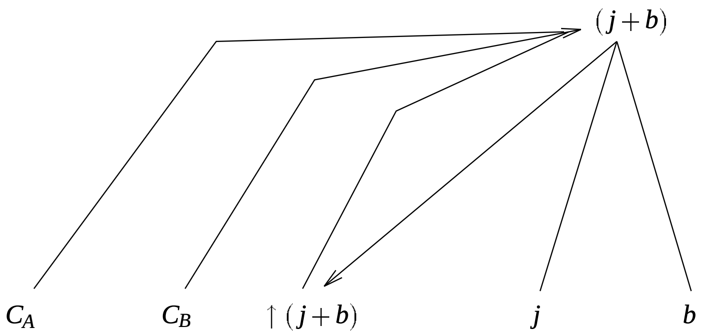


Figure 2.2: The Committees and Individuals Diagram.

(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

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

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 be(m)10.876(a)5.647(f)4.2

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

> 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

Ine fk omCof

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 whe

Anaphoric Processing			
Derivation	Constraint		
	Accessibility	Satisfiability	Resolution

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 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.

(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

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 1a)

eokde

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

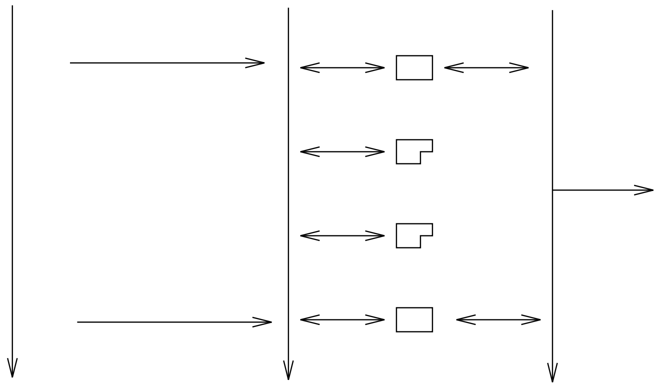
the informational entities for Jack and Jane together to form the appropriate referent for the interpretation of the anaphor *they*. Similarly, we need to collect the entities for the donkey and the horse that Jack and Jane own respectively. However, we do not wish or need to use the information concerning the ownership relationships between these people and their animals. They might beat each other's animals, for instance. For a **bound**

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

K

Cooper can not provide the bound inter-sentential anaphor-



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 i

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

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 r

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 $[\![$

(131)

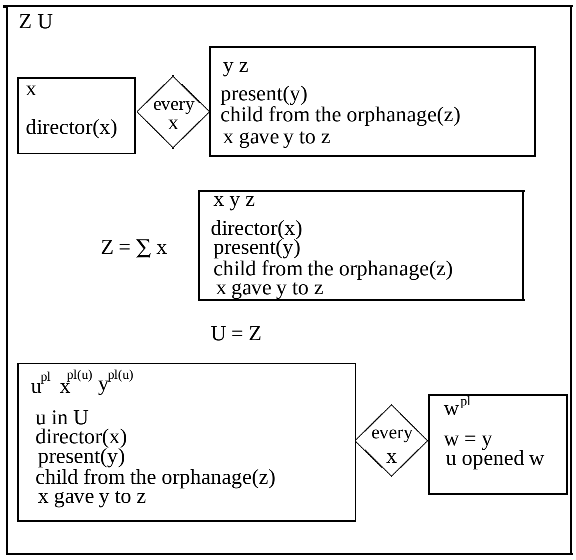
u v U Z
John(u)
Mary(v)
u saw v
$Z = u \oplus v$
U = Z
U went to the cinema

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

(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.

> ...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 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 $\forall\phi$

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 relatio

(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 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)ns t, whe ony

If $f(x)$

The prevalent theories can be split between the representat

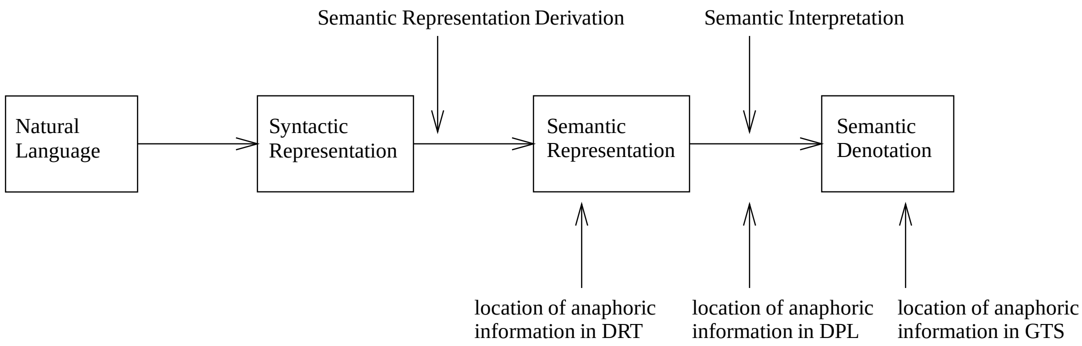


Figure 4.1: The components of a semantic theory and the location of anaphoric information within DRT, DPL and GTS.

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

4.1 The Representational Structures

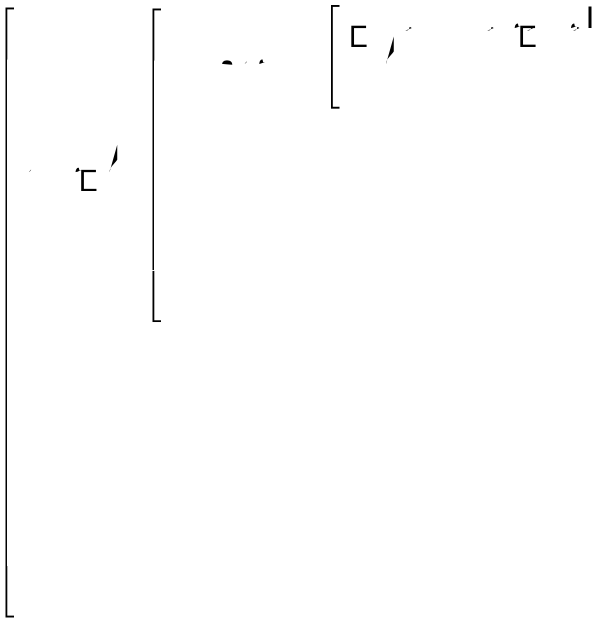
The semantic representations in GTS will be given as unificat

- 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 . For instance, a possible feature built after the analysis of the sentence in (168) is given in (169).

(168) Every farmer owns a donkey.

(169)



predicate. The particular features that can appear within the [illegible] complex feature will not be discussed in detail here. However, for the particular case of transitive verbal predicates, the [illegible] complex feature is split between [illegible], [illegible] and [illegible] features which contain features such as the following:

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 an-

(171) $M_1 = \langle D_1, F_1 \rangle$, where
$D_1 = \{a, b, c, d, e, f\}$,
$F_1(\llcorner \quad \llcorner) = \{a$

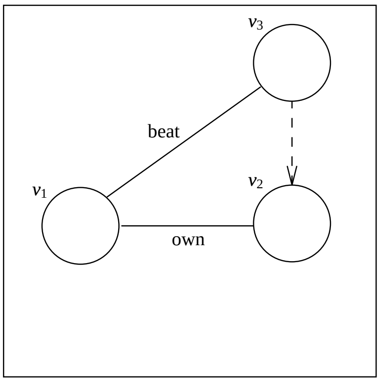


Figure 4.4: Denotation graph for *every farmer who owns a donkey beats it* with respect to a satisfying model.

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) [illegible]

The [illegible] feature contains appropriate syntactic information, while the [illegible] feature contains the semantic information. The [illegible] 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) [illegible] $_n$

```
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) [illegible]

(181) [illegible] $_{tv}$

```
Word owns: <cat> = v
```

```
   <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.

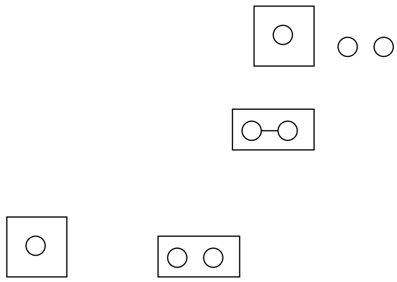
Rule {Nbar lexical noun}
```

```
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



(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

$$\left[\quad [\ [\ ]\ ]_{n} \right]_{det}$$

(187) A farmer

$$\left[\quad [\ [\ ]\ ]_{n} \right]_{det}$$

(188) No farmers

$$\left[\quad [\ [\ ]\ ]_{n} \right]_{det}$$

(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 and is to pass information up to the verbal predicate which will help determine the possible interpretation required. In particular, the feature specifies whether only one witness set must satisfy the verbal predicate or any number of witness sets. The feature 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 and . This requires the verbal relation to ensure that only one witness set of three farmers satisfies the predicate 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 verbal predicate. The semantic representation of ***no farmers***, given in (188), contains the features $-$ and $-$. 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)

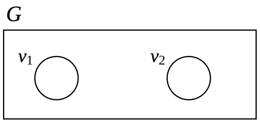


Figure 4.6: The graph derived after the arguments to the transitive verbal predicate given in (200) have been analysed.

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 [illegible] 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 [illegible] features.

As an example, the [illegible] 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 [illegible] feature is a positive polarity subject and object distributive reading with subject-wide scope and no uniqueness restriction.

(201) [illegible]

(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

[illegible]

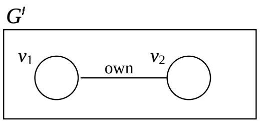


Figure 4.7: The graph derived from the analysis of (200).

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)

$$\left[\ \left[\ \right]\right]_{pro}$$

The feature variable will take the value of the particular pronoun, e.g., , etc. The feature variable will take the value either or while the feature 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*

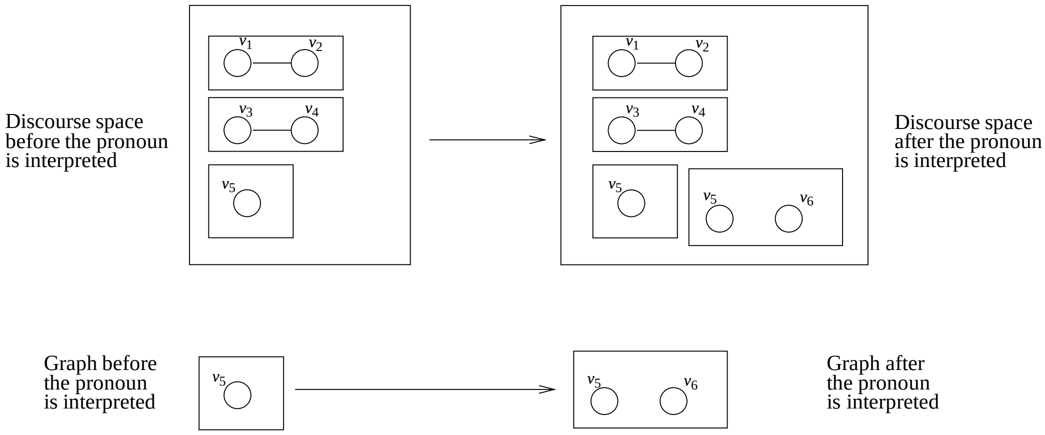


Figure 4.8: Interpretation of a referential pronoun.

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.

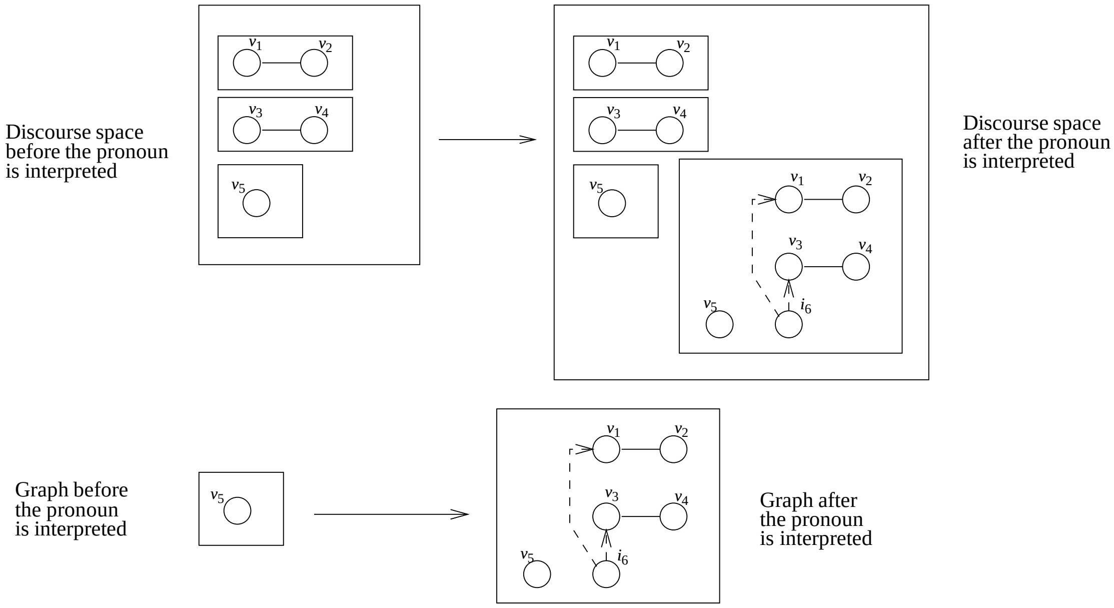


Figure 4.9: Interpretation of a bound pronoun.

(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 . 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 verbal relation for the donkey sentence in (208) is shown in (209).

(208) Every farmer who owns a donkec737(e)5w()4.10914(27-4.108)-6.931.

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.

sIf we determine all po00914(o)-6814 Tf18.2398 0 Tf.93181(s)-334.274(c)5.64311(a)5.64311(l)-10914(l)-6.9318

- The separation of lexical nouns, which derive zero-place pr

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

- $G[v_1'/v$

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 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.

$S_1, S_2, \ldots, S_n$ will produce a series of semantic representations $\alpha_1, \alpha_2, \ldots, \alpha_n$. The semantic interpretation function will be applied under the tempo914(n)-4.171

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 $\left[\ldots \left[\ldots \right] \right]_n \sqsubseteq \alpha$ and $\left(\ldots = \alpha/\langle \ldots \rangle \ \ldots = \alpha/\langle \ldots \rangle \right)$ then [illegible]

$[\![\alpha]\!]^{M,I} = \langle (G, \mathcal{D}), (i, G[v], \mathcal{D} \cup \{G[v]\}) \rangle$, 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 $\mathcal{C} = \begin{cases} \{X \subseteq F(\ldots) \mid |X| = 1\} & \text{If } \ldots \\ \{X \subseteq F(\ldots) \mid |X| \geq 1\} & \text{If } \ldots \end{cases}$ [illegible]

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.

$$\left[\ldots \left[\ldots \right] \right]_{det}$$

[illegible]

The [illegible] feature structure contains several features. The [illegible] feature contains the particular determiner predicate in question, for example [illegible] or [illegible]. The [illegible] feature determines the (syntactic) number. Finally, there are two features ([illegible] and [illegible]) 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 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).

If $\left[\quad \left[\quad A \quad \right] \right]_{det} \sqsubseteq \alpha$ and $\left(\begin{array}{l} = \alpha/\langle \quad \rangle \\ = \alpha/\langle \quad \rangle \\ A = \alpha/\langle \quad \rangle \end{array} \right)$ then

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

- $[\![A \quad]\!]^{M,I} = \langle (G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$
- $\langle i,\mathcal{C}\rangle \in G'$, $S = \bigcup_{X\in\mathcal{C}} X$

  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 $\mathcal{C}$.

- $\mathcal{C}' = \begin{cases} \{X \subseteq S \mid X = S\} & \text{If} \quad = \\ \{X \subseteq S \mid |X| \geq \frac{1}{2}|S|\} & \text{If} \quad = \\ \{X \subseteq S \mid |X| = 1\} & \text{If} \quad = \\ \{X \subseteq S \mid |X| = 1\} & \text{If} \quad = \\ \{X \subseteq S \mid |X| > 1\} & \text{If} \quad = \\ \{S - X \mid X = \{\}\} & \text{If} \quad = \\ \{S - X \mid X \subseteq S \wedge |X| < \frac{1}{2}|S|\} & \text{If} \quad = \\ \{X \subseteq S \mid |X| = 2\} & \text{If} \quad = \end{cases}$

  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 07TJ/R1280.24Tf100-1164.43.0235.459(w)9.46513(i)-6.9318j08e(00)TjTm

Feature	Possible Values
⟨ [illegible] ⟩,⟨ [illegible] ⟩, ⟨ [illegible] ⟩	+,−
⟨ [illegible] ⟩,⟨ [illegible] ⟩, ⟨ [illegible] ⟩	[illegible] ,| ,|

The [illegible] features have three values, [illegible] for sentence negation, | for verb phrase negation and | 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 ⟨ [illegible] ⟩ determines the scope of each noun phrase within the verbal reading. The possible values for this feature are given below.

Feature	Possible Values
⟨ [illegible] ⟩	[illegible] , [illegible]

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

(223) Every farmer owns a donkey.

For the verbal predicate [illegible] with a ⟨ [illegible] ⟩ feature set to [illegible] 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 ⟨

Features	No. donkeys owned by each farmer	No. donkeys owned by all farmers	Each donkey owned by all farmers
$\langle$ [illegible] $\rangle$: [illegible] $\langle$ [illegible] $\rangle : -$	≥ 1	≥ 1	not required
$\langle$ [illegible] $\rangle$: [illegible] $\langle$ [illegible] $\rangle : +$	1	≥ 1	not required
$\langle$ [illegible] $\rangle$: [illegible] $\langle$ [illegible] $\rangle : -$	≥ 1	≥ 1	required
$\langle$ [illegible] $\rangle$: [illegible] $\langle$ [illegible] $\rangle : +$	1	1	required

Table 5.1: Different readings of (223) due to values of $\langle$ [illegible] $\rangle$ and $\langle$ [illegible] $\rangle$.

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]

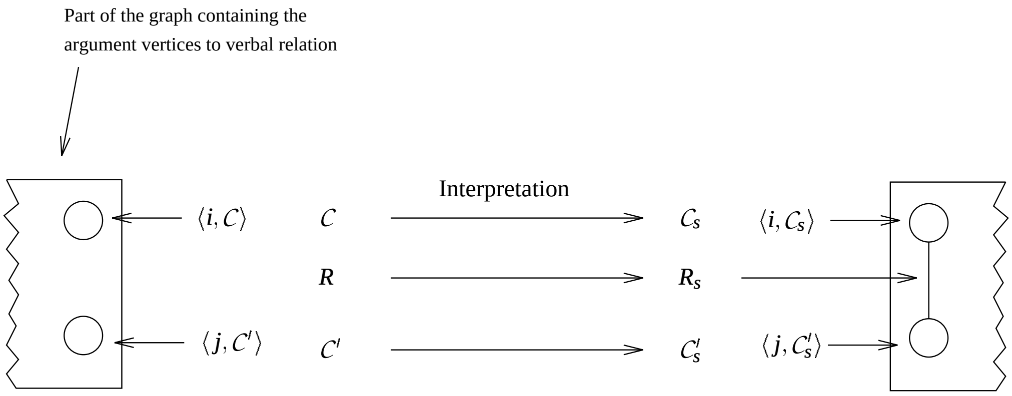


Figure 5.2: A viewpoint on the analysis of a transitive verbal predicate.

$\mathcal{C}'$ to satisfy the verbal reading. If any sets in $\mathcal{C}$ and $\mathcal{C}'$ 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 $\mathcal{C}$ and $\mathcal{C}'$ are related. That is, the situation is as shown in figure 5.2. From the denotation sets of the two arguments, $\mathcal{C}$ and $\mathcal{C}'$ we wish to construct new denotation sets $\mathcal{C}_s$ and $\mathcal{C}'_s$ which contain those sets from $\mathcal{C}$ and $\mathcal{C}'$, respectively, which satisfy the particular verbal reading. We will also derive a relation R_s specifying which elements (subsets of sets) in $\mathcal{C}_s$ and $\mathcal{C}'_s$ are related, as determined by the verbal reading. However, in viewing $\mathcal{C}_s$ as or control features, the ***subject rule*** or ***object rule*** derived is shown below, where the features and are meant to apply either to ⟨ ⟩ and ⟨ ⟩ or to ⟨ ⟩ and ⟨ ⟩.

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

The

(225) [illegible]

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

(226) $\lambda C_1, C_2, V.\exists S_1 \in$

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.

(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

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*** 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

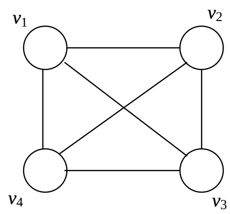


Figure 6.1: A graph.

used in the satisfaction of the verbal reading applied. This

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

- 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 $\mathcal{C}$, a semantic resolution function , and discourse space $\mathcal{D}$,
  if $[\![\alpha]\!]^{M,I,\mathcal{C}\ ,} = \langle(\langle\{\},\{\},\{\}\rangle,\mathcal{D})$

If

$$\left[\left[\; \right] \right]_{pro}$$

The feature contains three features. The feature specifies the particular pronominal predicate, e.g., , , . The feature specifies the type of pronoun, either or . The 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 is a set of vertex-graph pairs, then $_G$ is the set of graphs from .

The formal interpretation of referential and bound anaphors can now be directly given, where M is a model, I is a set of identifiers, $\mathcal{C}$ is an anaphoric constraint function, is an anaphoric resolution function, $\mathcal{D}$ is a discourse context and α is aa a

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

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$
- $(\mathcal{D}\ ,\mathcal{C}\ (\alpha,\mathcal{D},G)) = \langle\mathcal{C},\ \rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{C}$.
- $v = \langle i,\mathcal{C}\rangle$.
  The vertex for the anaphor is created.
- $A = \{\langle v,v_i\rangle|\langle v_i,G\rangle \in\ \}$ 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.
- $G_{t2} = \bigcup_{G\in\ _G} G$
  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 verbal predicate is treated.

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

In the analysis of the 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 $\langle C_s, C_o, R\rangle$ is derived by determining the interpretation rule, i.e., described by the information of the verbal predicate and taking the component-wise union of **all** triples $\langle C_t, C_t', R_t\rangle, C_t \subseteq$

- $R = \{\langle X, Y\rangle | \exists S_1 \in \mathcal{C}, \exists S_2 \in \mathcal{C}' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The relation R allows any pair of subsets from either argument.

- $R_a = \{\langle X, Y\rangle \in R | satis($

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.

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 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 $\mathcal{C}$ 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 defin

- : This feature is utilized in both the semantic interpretation of generalized quantifiers and verbal predicates. Shisene6137(e)5.64534(s)- inc8.78(g)-4.11137(e)5.6424.70R49 8(e)5.64

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

```
          <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
  ```

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>
  ```
  ```

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
$D_1 = \{a,b,c,d,e,f,g,h\}$,
$F_1(\quad) = \{a,b\}$
$F_1(\quad) = \{d,e,f,g\}$
$F_1(\quad) = \{\langle\{a\},\{d\}\rangle,\langle\{a\},\{e\}\rangle,\langle\{b\},\{f\}\rangle,\langle\{b\},\{g\}\rangle\}$
$F_1(\quad) = \{\langle\{a\},\{d\}\rangle,\langle\{a\},\{e\}\rangle,\langle\{b\},\{f\}\rangle,\langle\{b\},\{g\}\rangle\}$

I will assume a set of identifiers $I_1 = \{1,2,3...\}$. The anaphoric constraint function $\mathcal{C}$ as well as the anaphoric resolution function will not be specified. I will assume through the analysis that an appropriately realised set of $\mathcal{C}$ and 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) $\alpha_1 =$

The inductive procedure for the interpretation of a discourse was given in (253) and (254), repeated below.

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

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

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 . The interpretation of lexical nouns given in section 6.1.3 is repeated below.

If $\left[\quad , \left[\quad \right] \right]_n \sqsubseteq \alpha$ and $\left(= \alpha/\langle \quad \rangle \;\; = \alpha/\langle \quad \rangle \right)$ then $[\![\alpha]\!]^{M,I,C} \quad , \quad = \langle(G,\mathcal{D}),(i,G[v],\mathcal{D}\cup\{G[v]\})\rangle$, where,

- **[1]** $i \in I$

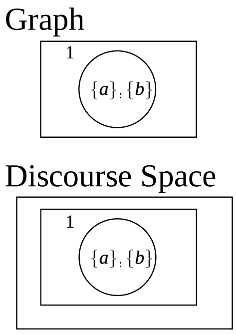
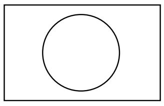
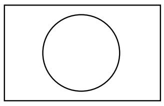


Figure 6.5: Output graph and discourse space after the analy





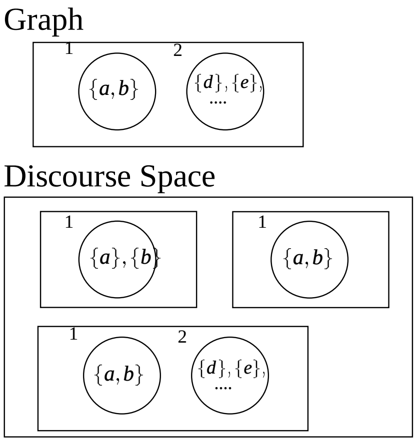


Figure 6.7: Output graph and discourse space after the analysis of the nominal predicate L in (268).

- 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 . The appropriate interpretation rule for determiner predicates is repeated below.

If $\left[\quad [\quad \begin{bmatrix} [& & A \\ & [& \\ & & D \end{bmatrix} \right]_{det} \sqsubseteq \alpha$ and $\left(\begin{array}{l} D = \alpha/\langle \quad [\quad [\quad \rangle \\ M = \alpha/\langle \quad [\quad [\rangle \\ A = \alpha/\langle \quad [\quad \rangle \end{array} \right)$ then

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

- **[1]** $[\![A \quad]\!]^{M,I,C} \quad , \quad = \langle (G, \mathcal{D}), (i, G', \mathcal{D}') \rangle$

- **[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]** $\mathcal{C}' = \begin{cases} \{X \subseteq S | X = S\} & \text{If } [illegible] = [illegible] \\ \{X \subseteq S | |X| \geq \frac{1}{2}|S|\} & \text{If } [illegible] = [illegible] \\ \{X \subseteq S | |X| = 1\} & \text{If } [illegible] = [illegible] \\ \{X \subseteq S | |X| = 1\} & \text{If } [illegible] = [illegible] \\ \{X \subseteq S | |X| > 1\} & \text{If } [illegible] = [illegible] \\ \{X \subseteq S | X = S\} & \text{If } [illegible] = [illegible] \\ \{X \subseteq S | |X| \geq \frac{1}{2}|S|\} & \text{If } [illegible] = [illegible] \\ \{X \subseteq S | |X| = 2\} & \text{If } [illegible] = [illegible] \end{cases}$

  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.

- [illegible] = [illegible]
- [illegible] = [illegible]
- 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*
- $i = 2$
- $\mathcal{C} = \wp\{d, e, f, g\}$

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 $\mathcal{C}'$ is as shown below.

- $\mathcal{C}' = \{\{d, e\}, \{d, f\}, \{e, f\}, \{d, g\}, \{e, g\}, \{f, g\}, \{d, e, f\}, \{d, e, g\}, \{d, f, g\}, \{e, f, g\}, \{d, e, f, g\}\}$

The newly derived graph G''

- **[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*

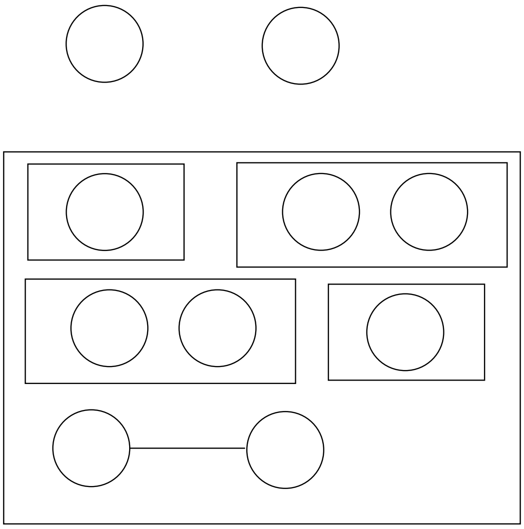
In **[3]**, the two vertices describing the arguments are extracted. The values of the vertex structures are given below.

- $i_1 = 1$
- $\mathcal{C} = \{\{a, b\}\}$
- $i_2 = 2$
- $\mathcal{C}' = \{\{d, e\}, \{d, f\}, \{e, f\}, \{d, g\}, \{e, g\}, \{f, g\}, \{d, e, f\}, \{d, e, g\}, \{d, f, g\}, \{e, f, g\}, \{d, e, f, g\}\}$

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.

- $R = \{\langle \{a\}, \{d\} \rangle, \langle \{a\}, \{e\} \rangle, \langle \{a\}, \{f\} \rangle, \langle \{b\}, \{d\} \rangle, \langle \{b\}, \{e\} \rangle, \langle \{b\}, \{f\} \rangle, \langle \{a, b\}, \{d\} \rangle, \langle \{a, b\}, \{e\} \rangle, \langle \{a, b\}, \{f\} \rangle, ...\}$

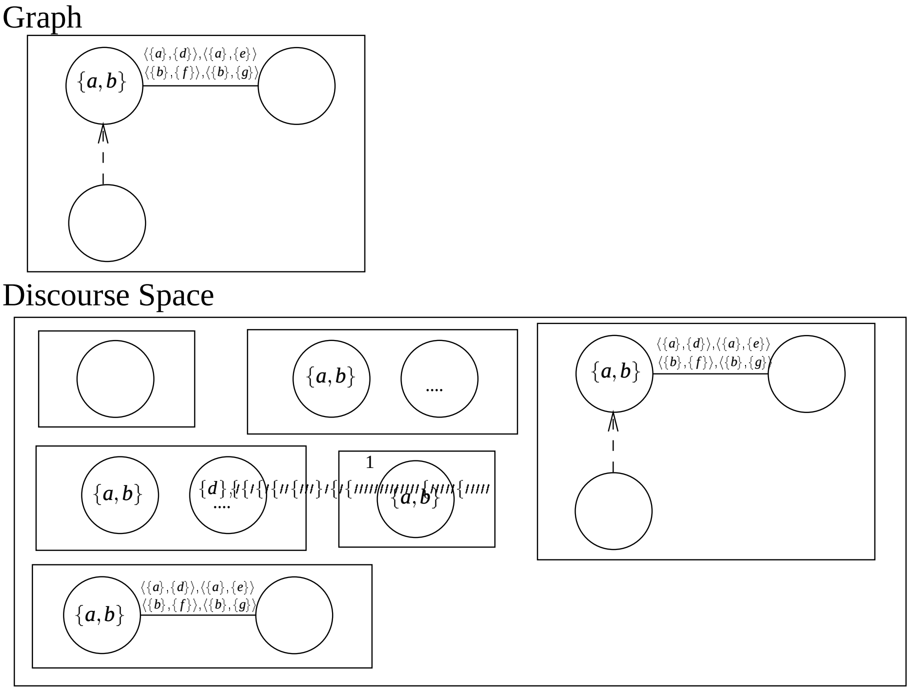
In **[5]** we limit this relation to onlyn



If $\left[\quad \left[\quad \right] \right]_{pro} \sqsubseteq \alpha$ then

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

- **[1]** $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$
- **[2]** $(\mathcal{D}\ , \mathcal{C}\ (\alpha, \mathcal{D}, G)) = \langle \mathcal{C},\ \rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph


Graph
{a,b}
⟨{a},{d}⟩,⟨{a},{e}⟩
⟨{b},{f}⟩,⟨{b},{g}⟩
Discourse Space
{a,b}
....
{a,b}
....
1
{a,b}
{a,b}
⟨{a},{d}⟩,⟨{a},{e}⟩
⟨{b},{f}⟩,⟨{b},{g}⟩
{a,b}
⟨{a},{d}⟩,⟨{a},{e}⟩
⟨{b},{f}⟩,⟨{b},{g}⟩

- **[2]** $(\mathcal{D}\ ,\mathcal{C}\ \ (\alpha,\mathcal{D},G)) = \langle \mathcal{C},\ \ \rangle$
  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{C}$.

- **[3]** $v = \langle i,\mathcal{C}\rangle$.
  The vertex for the anaphor is created.

- **[4]** $A = \{\langle v,v_i\rangle | \langle v_i,G\rangle \in\ \ \}$ 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.

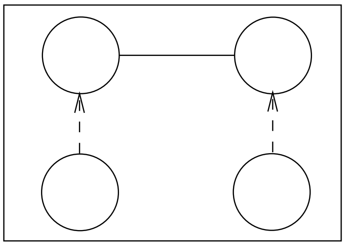
- **[5]** $G_{t2} = \bigcup_{G\in\ \ _G} G$
  A graph G_{t2} is created from the union of the antecedent graphs.

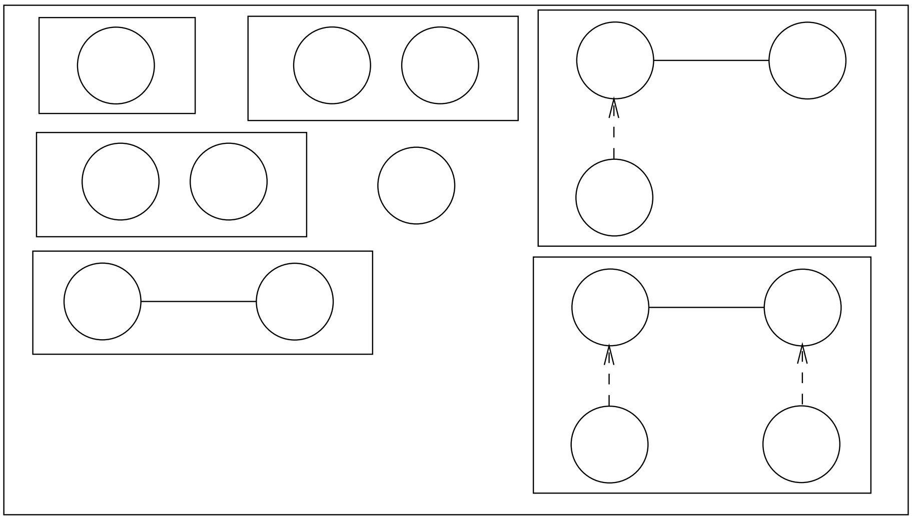
- **[6]** $G' = G[v] \cup G_{t1} \cup G_{t2}$
  The graph for the anaphor is the union of the extension of the graph G with the anaphor vertex along with the graphs G_{t1} and G_{t2}.

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 senten

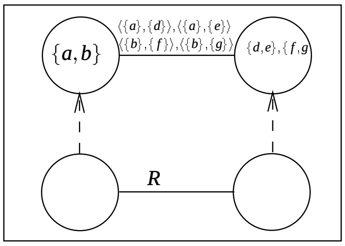




If $\left[\ldots \right]_{tv} \sqsubseteq \alpha$, and $\left(\begin{array}{l} = \alpha/\langle \ldots \rangle \\ \mathsf{A} \quad = \alpha/\langle \ldots \rangle \\ \mathsf{A} \quad 2 = \alpha/\langle \ldots 2 \rangle \\ \quad = \alpha/\langle \ldots \rangle \\ \mathsf{AA} \quad = \alpha/\langle \ldots \rangle \end{array} \right)$ then,

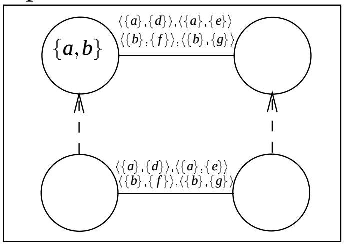
$[\![\alpha]\!]^{M,I,C} \ldots = \langle (G_1, \mathcal{D}_1), (i_1, G_4, \mathcal{D}_3 \cup \{G_4\}) \rangle$ where

- **[1]** $[\![\mathsf{A} \quad]\!]^{M,I,C} \ldots = \langle (G_1, \mathcal{D}_1), (i_1, G_2, \mathcal{D}_2) \rangle$
- **[2]** $[\![\mathsf{A} \quad 2]\!]^{M,I,C} \ldots = \langle (G_2, \mathcal{D}_2), (i_2, G_3, \mathcal{D}_3) \rangle$
- **[3]** $v = \langle i_1, C \rangle$ where $\langle i_1, C \rangle \in G_3$ and $v' = \langle i_2, C' \rangle$ where $\langle i_2, C' \rangle \in G_3$
  The vertices for each argument are determined via the identifiers i_1 and i_2.
- **[4]** $R = \{\langle X, Y \rangle | \exists S_1 \in C, \exists S_2 \in C' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The relation R allows any pair of subsets from either argument.
- **[5]** $R_a = \{\langle X, Y \rangle \in R | satis(G_3[\langle v, v', R \rangle], \quad) \wedge \{\langle v, X \rangle, \langle v', Y \rangle\} \subseteq \quad \}$
  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 then:
  - If $\quad = \ldots$, $\phi' = (((\phi(C)), (C')), (F(V)))$
  - If $\quad = \ldots$, $\phi' = (((\phi(C')), (C)), (F($

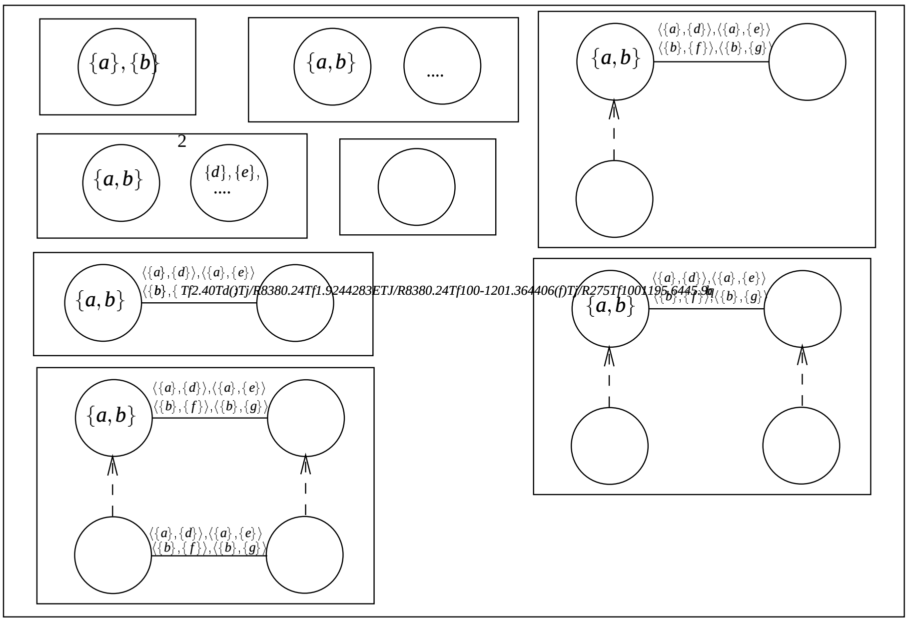

{a,b}
⟨{a},{d}⟩,⟨{a},{e}⟩
⟨{b},{f}⟩,⟨{b},{g}⟩
{d,e},{f,g
R

In **[6]** we create the appropriate verbal rule which is identical to the rule created for the analysis of the verbal predicate

Graph



Discourse Space



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 ([illegible]) 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.

(2785.64534(-)4.23177]TJ11137(r)-324i6m.9307(n9(,)-232.199(o)-4.10714(h)-4.11131(o)-4.1091(a)5.64422(p)-4.
iety

I14(4)-4.10915())-477.9]TTne236.9064ane

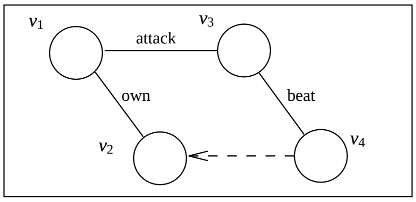


Figure 6.17: The final graph describing the sentence in (280).

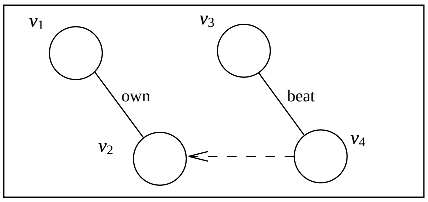


Figure 6.18: Graph derived during the analysis of the sentence in (280) just after the analysis of the verbal predicate.

(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) $\exists A \in \mathcal{C} : \exists B \in \mathcal{C}' : [\forall x \in A \exists y \in B : \langle \{x\}, \{y\} \rangle \in F(\quad)] \wedge [\forall y \in B \exists x \in A : \langle \{x\}, \{y\} \rangle \in F(\quad)]$

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.

(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*

(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 [illegible]

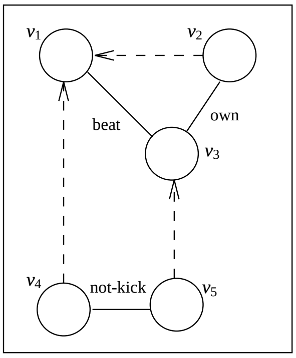


Figure 6.19: The denotation graph for the discourse in (300) within a satisfying model.

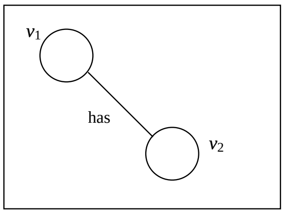


Figure 6.20: The denotation graph for the noun phrase *every man who has a dime* in (301) with respect to a satisfying model.

of anaphoric information are straightforwardly handled by the threading of the discourse space though an interpretation of a discourse.

object argument to the verbal predicate [illegible]. 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***

Chapter 7

Computational Issues

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 $O(p(n))$

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, $\mathcal{C}$ is an anaphoric constraint function, 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) $[\![\alpha]\!]^{M,I,\mathcal{C}\;\;,} = \langle(G,\mathcal{D}),(i,G',\mathcal{D}')\rangle$

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

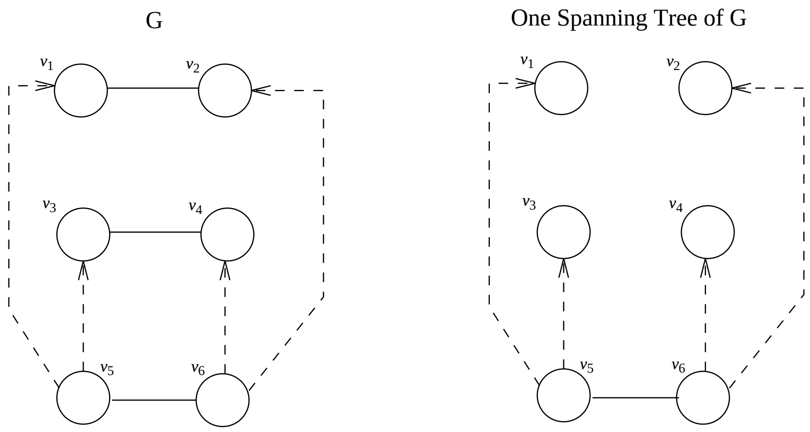
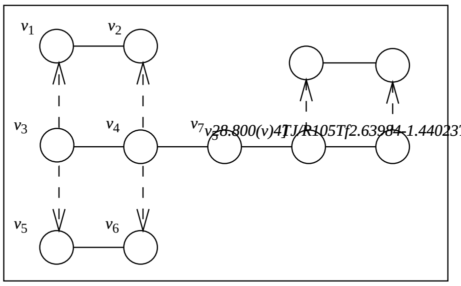


Figure 7.1: A graph with associated spanning tree

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 boun(o)17.8075utht


v_1
v_2
v_3
v_4
v_7
v_5
v_6

O(el) where e is the number of edges and l

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

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 $_0, _1, \ldots, _n$, where $_i (1 \leq i \leq n)$ is derived by modifying $_{i-1}$

```
[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]] ] ].
```

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 [illegible] 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 ⟨[illegible]⟩ 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:`VfAs4622(.)where
  ```

```
[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 [illegible] 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]], [[f5f5f5f5f
```

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 C and functions from the framework.

```
Choose antecedents by giving vertex-graph pairs of the :
```

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.

```
---------------------------------------------
```

```
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 t straint 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 h.52048(s)-5.52048(i)-6

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 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 a

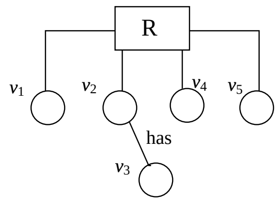


Figure 8.1: The denotation graph describing (310) with respect to a suitable model.

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

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.

Cooper, R. 1979. The Interpretation of Pronouns. *Pages 61–92 of:* Heny, F., & Schnelle, H. (eds), *Syntax and Semantics 10: Selections from the Third Groninge*

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

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*

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, $\mathcal{C}$ is an anaphoric constraint function, is an anaphoric resolution function and α is a feature-based semantic representation.

If $\left[\quad \right]_n \sqsubseteq \alpha$ and $\left(\begin{array}{l} = \alpha/\langle \quad \rangle \\ = \alpha/\langle \quad \rangle \end{array} \right)$ then

[[

- $\mathcal{C}$

- $G' = G[\langle i, \mathcal{C} \rangle]$

  Create the graph for the anaphor by extending G with the anaphor vertex $\langle i, \mathcal{C} \rangle$.

If $\left[\quad \left[\quad \right] \right]_{pro} \sqsubseteq \alpha$ then

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

- $i \in I$ is an identifier not so far used in any vertex in any graph in $\mathcal{D}$

- $(\mathcal{D}, \mathcal{C}\ (\alpha, \mathcal{D}, G)) = \langle \mathcal{C}, \ \rangle$

  Obtain the anaphor denotation set and antecedent vertex-graph pairs by applying the anaphoric resolution function to the current discourse context and the set of anaphor antecedent denotation pairs provided by the anaphoric constraint function $\mathcal{C}$.

- $v = \langle i, \mathcal{C} \rangle$.

  The vertex for the anaphor is created.

- $A = \{\langle v, v_i \rangle | \langle v_i, G \rangle \in \ \}$ 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.

- $G_{t2} = \bigcup_{G \in \ _G} G$

  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, $\mathcal{C}$

- $[\![\mathsf{A} \quad 2]\!]^{M,I,C} \quad , \quad = \langle (G_2, \mathcal{D}_2), (i_2, G_3, \mathcal{D}_3) \rangle$
- $v = \langle i_1, C \rangle$ where $\langle i_1, C \rangle \in G_3$ and $v' = \langle i_2, C' \rangle$ where $\langle i_2, C' \rangle \in G_3$
  The vertices for each argument are determined via the identifiers i_1 and i_2.
- $R = \{\langle X, Y \rangle | \exists S_1 \in C, \exists S_2 \in C' : X \subseteq S_1 \wedge Y \subseteq S_2\}$
  The relation R allows any pair of subsets from either argument.
- $R_a = \{\langle X, Y \rangle \in R | satis(G_3[\langle v, v', R \rangle], \quad) \wedge \{\langle v, X \rangle, \langle v', Y \rangle\} \subseteq \quad \}$
  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 then:
  - If $=$, $\phi' = (((\phi(C)), (C')), (F(V)))$
  - If $=$, $\phi' = (((\phi(C')), (C)), (F(V)))$

  The verbal reading rule is derived, the arguments to ϕ being given in an order determined by the feature .
- There is a mapping from $\langle C, C', R_a \rangle$ to $\langle C$

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

```
    (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,
```

```
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.
```

```
---------------------------------------------
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
```