Technical Report

UCAM-CL-TR-237
ISSN 1476-2986

Number 237

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

Computing presuppositions in an incremantal language processing system

Derek G. Bridge

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted
April 1991 by the author for the degree of Doctor of
Philosophy to the University of Cambridge, Wolfson College.

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Summary

This thesis describes the design and implementation of a natural language analysis system for the computation of *presuppositions.* The system is one in which syntactic, semantic and pragmatic processing are interleaved with feedback to syntactic analysis from semantic and pragmatic processing. The thesis begins by illustrating how the system processes definite noun phrases. The mechanisms used for this are then shown to be easily extensible to processing other parts of speech such as indefinite noun phrases and verb phrases.

Definite noun phrases have been said to be presupposition triggers. This means that traditionally they have been seen as licensing certain inferences — presuppositions. In the system described herein, presuppositions are treated as a special kind of inference: *preconditions.* This treatment for definite noun phrases can be extended to give a uniform account of all presupposition triggers (e.g. factive verbs). It is a view that makes it clear that presuppositions are not 'optional extras' that might or might not be derived once a semantic representation of an utterance has been produced. Rather, they play an essential role in driving the utterance analysis process: the failure of a presupposition, i.e. failure to satisfy a precondition, can direct the system to choose an alternative reading of an utterance of an ambiguous sentence.

As it processes an utterance, the system builds and regularly consults a representation of contextual knowledge referred to as a *discourse model.* Importantly, the system checks whether presuppositions are satisfied against the discourse model. Presupposition failure, i.e. a presupposition not being satisfied by the discourse model, is not necessarily the same as a presupposition being false in, e.g., the 'real' world. Checking presuppositions for satisfaction in a discourse model and not for truth in a possible world offers new ideas on the behaviour of presuppositions in utterances of negative and complex sentences.

In utterances of negative sentences, presuppositions must still be satisfied by the discourse model. Presuppositions cannot be cancelled as they can in other accounts. Rather, presupposition "cancellation" data is explained in terms of utterances that make metalinguistic statements about the model-theoretic interpretation of the discourse model. It is also shown that computing presuppositions in an incremental system gives a simple account of most of the data relating to the behaviour of presuppositions in utterances of compound sentences and longer stretches of text (the so-called "projection problem"). Presuppositions must again be satisfied by the discourse model, but they may be satisfied by virtue of changes made to the discourse model by earlier parts of the utterance or text.

Contents

Chapter 1

Introduction

This thesis investigates *presupposition* in computational linguistics. In the past, presupposition in natural language processing has mostly been viewed as a type of *inference* drawn from natural language sentence or utterance readings. Within this framework, many different conceptions of presupposition have been formulated. Speaking loosely in order to embrace these formulations, what distinguishes presuppositions from other inferences is that they are, in some way, background assumptions against which an utterance is made. How to capture this more precisely has long been a source of controversy.

This dissertation presents a reformulation of presupposition. The status of presuppositions as background assumptions is retained, but presuppositions are no longer viewed as 'ordinary' inferences. The tack taken in this work is that presuppositions are a special kind of inference, namely those that are *preconditions.* Where presuppositions have been viewed as preconditions before, they have mostly been presented as preconditions of *speech acts.* In the cases where, as here, they have been presented as preconditions of *utterances*, or more strictly, of utterance readings (as in Strawson's work), they have been seen as propositions that must be satisfied in a model (in the model-theoretic sense). In my reformulation, presuppositions need only be satisfied by a discourse model, which is a representation of contextual knowledge.

Viewing presuppositions as preconditions gives rise to a different treatment of the computation of presuppositions in automatic natural language analysis systems. *Incremental* processing of utterances, where the application of different types of knowledge is interleaved and there is feedback between the different modules of the design, furnishes a good foundation for this. An analysis of an utterance is constructed in a cumulative fashion, working from left-to-right through the utterance. This enables the discourse model to be built in such a way as to represent the way context is augmented as discourse processing proceeds.

The first half of this thesis describes an incremental language understanding system of this kind. The second half shows how my account of presuppositions is easily and naturally incorporated into the system and looks at the data that has confounded other approaches to presupposition. This includes accounting for the behaviour of presuppositions in utterances of negative and complex sentences.

1.1 Thesis Overview

Chapter 2 looks at types of natural language analysis systems and their basic designs. It says what it means for a system to be one that processes utterances in an incremental fashion and looks at the possible advantages of incremental processing. It concludes with brief overviews of two other incremental systems.

Chapter 3 describes T42, the incremental language processing system which I have developed using Haddock's work on such systems [Haddock 1987a, 1987b] as a starting point.[1] The principles of T42 are highlighted, most notably its integration of syntactic, semantic and pragmatic processing, and each of its modules is described in detail.

Chapter 4 shows how the modules of the system work together by giving an extended example. The example, again following Haddock, is one of prepositional phrase attachment decisions. Attractive features of the system are highlighted, including the way that Haddock proposes the uniqueness constraint on definite noun phrases should be handled.

Chapter 5 discusses the principles underlying T42's processing of definite noun phrases. Their processing is central to the later parts of the thesis which deal with presuppositions since definite noun phrases are one particular presupposition trigger: processing of other presupposition triggers will be uniform with definite noun phrase processing. The chapter looks at "referential" and "attributive" uses of definite noun phrases. It suggests that the attributive use is not as homogeneous as has been implied by other work and relates the referential use, and T42's approach to it, to Strawson's treatment of definite reference as presupposition.

Chapter 6 demonstrates the extensibility of T42: a variety of other parts of speech are shown to be amenable to a treatment which retains the principles described in Chapters 3, 4 and 5. The extensions emphasise the idea of making use of a discourse model to get appropriately 'fleshed out' representations of utterance readings. The chapter also develops new definitions (albeit technical definitions based on the way that T42 works) of "given" and "new", which rely on a distinction between demanding that entities be in the discourse model and demanding that properties of entities be in the discourse model. It also compares T42 to Discourse Representation Theory (DRT) and suggests that T42 has an ability that DRT does not: it has structures that can maintain a record of referential ambiguity.

Chapter 7 elaborates my characterisation of presuppositions and shows that T42 can handle other presuppositions in the way that it handles definite noun phrases. This characterisation is compared to other definitions that can be found in the linguistics literature. In this chapter, the focus is on the

[1] T42 should be pronounced Tee-Forty-two and not tea-for-two. This name comes from the number of the office in which system development was carried out.

presuppositions of simple utterances; negative and compound utterances are dealt with later. The chapter concludes with a look at the issues surrounding presupposition failure.

In Chapter 8, the interaction of presuppositions with negation is described both in terms of previous approaches and of my own approach and its realisation in T42. My account treats negation as semantically unambiguous, i.e. I use narrow-scope negation which preserves all presuppositions. This would seem to lead to problems in accounting for so-called presupposition "cancellation"; this is the data which has motivated alternative accounts of negation (e.g. ambiguous negation). I suggest that the resolution of this problem lies again in drawing the distinction between presupposition satisfaction by the discourse model and presuppositions being true in a model. Presupposition "cancellation" is explained as a metalinguistic use of language commenting on the interpretation of discourse model propositions.

Chapter 9 looks at how to determine the presuppositions of complex utterances (the so-called "projection problem"). T42's approach is presented with a comprehensive set of examples. The approach relies on processing utterances from left-to-right: earlier parts of a complex utterance or text change the discourse model so that subsequent presuppositions can be satisfied. It is shown that T42 can handle naturally many cases that have previously caused problems, although there does remain a set of examples that are not fully accounted for.

Chapter 10 reviews alternative approaches to the projection problem, showing their weaknesses and the data they fail to account for. It concentrates on computational approaches and shows that most of the problems they have stem from treating presuppositions as ordinary inferences rather than treating them as inferences that are more specifically preconditions.

Finally Chapter 11 presents conclusions and suggestions for future work.

The rest of the present chapter contains further introductory material. Section 1.2 covers typographic conventions used in the rest of the text. Section 1.3 contains definitions of some of the terminology used in the thesis. In particular it gives rough characterisations of terms such as "utterance", "semantics" and "pragmatics", as used in this thesis. Section 1.4 covers different types of inferences such as "entailments" and "implicatures". Section 1.5 gives an overview of T42 and Section 1.6 looks at the problems that must be dealt with in computing presuppositions.

1.2 Conventions

This dissertation deals with written and not spoken communication. However, in line with common usage, I will refer to the conversational participants as the "speaker" and the "hearer". The speaker will be the user of the natural language processing system, typing utterances at a keyboard for processing by the hearer, i.e. by the computer system. As a convention, I will take the speaker (user) to be female and the hearer (system) to be male. This allows the conversational participants to be distinguished even when using pronominal reference and gives a simple and consistent abbreviation scheme: 'S' – Speaker/She; 'H' – Hearer/He.

The following typographic conventions, covering both font type and form of quotation marks, are used in examples:

"This is a sentence."
"*This is an utterance.*"
'This is a proposition.'

Propositions will also be shown using first-order predicate calculus (e.g. `FAT(Falstaff)` ∧ `BALD(Falstaff)`) and may also be given in a combination of English and logical connectives (e.g. (Falstaff is fat) ∧ (Falstaff is bald)). Propositions can represent sentence or utterance readings. Definitions of sentence and utterance are given in the next section.

The following symbols will also be used (following [Levinson 1983]):

?? – This signals a pragmatically anomalous sentence or utterance.

* – This signals a syntactically ill-formed or semantically anomalous sentence or utterance.

? – This signals an anomaly but is non-committal about its nature: the nature of the infelicity will be explained in the text surrounding the example.

In examples, I will often show an utterance and the inferences (including those that are preconditions) that may be drawn from it. I show only the inferences that are important to my purposes at that point: in all cases, there might be other inferences that will not be shown.

1.3 Sentences and Utterances

Before going further, I will make some of the terminology used in this thesis a little clearer. I would not want to claim that I am giving definitions here. What follows are rough characterisations of the most important terms.

A *sentence* will simply be a well-formed string of words of the language. An *utterance of a sentence* (or, simply, *utterance*) will be a pair comprising a sentence and a context; it is an instance of the sentence, used on a particular occasion, in a particular context [Gazdar 1979, p.4].

The meanings of sentences and utterances will be captured by the notions of *sentence readings* and *utterance readings*. A reading can be thought of as a *proposition* which will, if we assume a bivalent semantics, be either true or false with respect to a particular model. Since sentences and utterances may be ambiguous, they are associated with *sets* of readings.

Sentence readings are the realm of *semantics*, which I take to account for the meaning of sentences, devoid of any context (context-independent meaning). More specifically, semantics captures the truth-conditions of a sentence. Since this is context-independent it cannot make use of speaker intentions and is thus restricted to 'literal' meaning.

Utterance readings involve both semantics and *pragmatics.* Stalnaker says that pragmatics has two major functions: "...first, to define interesting types of speech acts and speech products; second, to characterize the features of the speech context which help to determine which proposition is expressed by a given sentence. The analysis of illocutionary acts is an example of a problem of the first kind; the study of indexical expressions is an example of the second." [Stalnaker 1972, p.383]. This is a useful division: this thesis is concerned almost exclusively with problems of the second kind. One way of viewing this is that pragmatics takes the 'partial' meanings supplied by semantics and augments, modifies (e.g. 'fills in' indexicals) or rules some of them out using contextual knowledge. In other words, following [Stalnaker 1972], the propositions that represent utterance readings, which we wish to assess for truth in a possible world, are partially determined by semantics and partially determined by pragmatics with reference to the context (what I refer to in T42 as the discourse model).

I will not supply definitions of "context" or "discourse model" until Section 3.5 of Chapter 3 where I give them 'definitions' in terms of the knowledge which T42 deploys in utterance processing. In other words, my definitions are technical ones. This seems more appropriate for otherwise very nebulous concepts such as context and discourse model. For now I leave these terms as intuitive notions.

I will also refer to positive and negative sentences and to simple and compound sentences. These are little more than convenient 'labels' to facilitate the exposition. Roughly, a *positive sentence* will be one that does not contain the word "not" (e.g. "It is likely that Shakespeare wrote *Macbeth.*"). I shall not deal with cases where negation is incorporated morphologically (e.g. "It is unlikely that Shakespeare wrote *Macbeth.*). Sentences containing the word "not" will be referred to as *negative sentences* (e.g. "It is not likely that Shakespeare wrote *Macbeth.*"). This still leaves cases such as: "Bacon did not write *Macbeth*, but he helped Shakespeare with the ending.". This is best labelled as a positive conjoined with a negative.

A *simple sentence* will be a single clause, i.e. a clause with no constituent clauses in it (e.g. "Shakespeare wrote *Macbeth.*"). A *complex* or *compound sentence* will be one with sentential clauses in it (e.g. "It is likely that Shakespeare wrote *Macbeth.*", "Shakespeare wrote *Macbeth* and he wrote *Hamlet.*").

I will also use the phrases *positive utterance, negative utterance, simple utterance* and *complex utterance* (or *compound utterance*) as abbreviations for *utterance of a positive sentence, utterance of a negative sentence*, etc.

1.4 Types of Inference

Since I will be drawing a distinction between inferences that are preconditions and other inferences in this thesis, it is useful to review the main types of linguistic inference that have been identified in the linguistics literature, and to show the ways in which they differ from each other. I will cover entailments, presuppositions (as inferences that are not preconditions), conversational implicatures and conventional implicatures. I will give brief descriptions of each but since they are not really distinguishable by their form, I will conclude with a short comparison of their distinguishing behaviour. There are many problems with these definitions, particularly with those for "implicatures". I acknowledge this but I have not drawn these problems out in the text. Since, on the whole, these concepts are not germane to my own work but are needed when explaining the work of others, giving better definitions has not been of paramount importance.

I emphasise again that this section does not treat presuppositions as preconditions.

1.4.1 Entailment

Entailments are *logically valid* inferences. The relationship of entailment holds between propositions:

> A proposition P_1 entails a proposition P_2 iff whenever P_1 is true, P_2 must also be true (i.e. in all worlds in which P_1 is true, P_2 is true).

Strictly, sentences and utterances do not have entailments except derivatively, in the sense that their readings, i.e. the propositions they convey, do have entailments. Two simple examples are (where proposition (a) entails proposition (b)):

(1) a. 'Lear is a king.'
b. 'Lear is male.'

(2) a. 'The fool has five teeth missing.'
b. 'The fool has a tooth missing.'

1.4.2 Presupposition

Since I am discussing the work of a number of other people here, I am reluctant to give a definition of *presupposition* at this stage: there have been so many different attempts.[2] There is even disagreement about whether presupposition is a relation between sentences and propositions, between utterances and propositions or between speech acts and propositions. For now, a simple *test* for presupposition will have to stand in place of a definition:

> If sentence S expresses proposition P_1 and P_1 entails P and the negation of S has a reading P_2 which also entails P, then S presupposes P

[2] For my own attempt, see Chapter 7.

In the examples, (a) is the sentence, (b) is its negation and (in at least some theories) (c) is a presupposition of (a) and also a presupposition of (b) (or one of the readings of (b)):

(3) a. "The Prince of Denmark is mad."
b. "The Prince of Denmark isn't mad."
c. 'There is a Prince of Denmark.'

(4) a. "Lear regrets that he hit the fool."
b. "Lear doesn't regret that he hit the fool."
c. 'Lear hit the fool.'

I shall say more about the problems of presuppositions in Section 1.6, after introducing my system, T42, in Section 1.5.

1.4.3 Conversational Implicature

Grice proposed certain maxims of coöperative interaction which allow a hearer to infer that a speaker intends to convey meaning additional to the truth-functional content of her utterance, this extra meaning being meaning that is "conversationally implicated" [Grice 1975, 1978]. The maxims are invoked as part of following a general principle of coöperation:

> **The Coöperative Principle**
> Make your contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.

The maxims Grice proposed were those of Quality (roughly 'Be truthful'), Quantity (roughly 'Be as informative as is required'), Relation (roughly 'Be relevant') and Manner (roughly 'Be perspicuous'). Grice pointed out that the list of maxims was probably not complete nor free from redundancy. Many authors have tried to 'tighten up' this Gricean framework by introducing formalism or new maxims (e.g. [Kempson 1975], [Gazdar 1979], [Joshi *et al* 1984], [Atlas & Levinson 1981] and [Hirschberg 1985]).

In the simple case, the speaker intends, and the hearer infers, a *conversational implicature* which amplifies the speaker's utterance. This is a case where an implicature arises due to a participant *observing* the maxims[3]. Implicatures may also arise when a speaker blatantly violates a maxim. At first blush, it might appear that she is being uncoöperative, but the blatancy of the violation will be an indication to the hearer that the assumption of coöperation is preserved by assuming that further information is intended. Such blatant violations of maxims giving rise to implicatures, Grice termed *floutings*.

Grice drew a distinction between *particularised* and *generalised conversational* implicatures. Particularised implicatures depend upon some special

[3] The term "observing" comes from [Levinson 1983, p.104]. There is some question as to whether Grice thought that implicatures could arise in this way.

feature of the context of utterance: the same sentence uttered on a different occasion (i.e. in a new context) where the special feature is missing from context or some other feature is present in context might not license the implicature. Generalised implicatures, on the other hand, tend to be constant over contexts *unless some extra blocking feature is present*, i.e. if the same sentence is uttered in many different contexts, the implicature will still be understood unless the speaker explicitly blocks it or relies on the context to block it.

Here are some examples of generalised conversational implicatures which will be useful in later parts of this thesis. They are explained in the subsequent text.[4]

(5) a. "*Richard III jumped on his horse and he rode into the sunset.*"
b. 'Richard III jumped on his horse *and then* he rode into the sunset.'

(6) Scale: <'some', 'all'>[5]
a. "*Some of the fairies like Oberon.*"
b. 'Not all of the fairies like Oberon'

(7) Scale: <'one', 'two', 'three',...>
a. "*Macbeth met three witches.*"
b. 'Macbeth met no more than three witches'

(8) a. "*Sir Andrew is a knight or a knave.*"
b. 'Sir Andrew may be a knight', 'Sir Andrew may not be a knight', 'Sir Andrew may be a knave', 'Sir Andrew may not be a knave'[6]

(9) a. "*If Sir Toby is drunk, he will awaken Olivia.*"
b. 'Sir Toby may be drunk', 'Sir Toby may not be drunk', 'Sir Toby may awaken Olivia', 'Sir Toby may not awaken Olivia'

In these examples, (5b) is a possible generalised conversational implicature of (5a) arising from observing the maxim of Manner: in the absence of any information to the contrary, the hearer, H, can assume the speaker, S, is being orderly and there is some temporal ordering on the conjuncts. (6b) and (7b) are generalised conversational implicatures of (6a) and (7a), arising in connection with the maxim of Quantity. They are referred to as *scalar implicatures.* They arise as follows: a speaker using an utterance which contains an expression for which there is some other expression of roughly similar length and familiarity but which is stronger in its meaning conveys that she is not in a position to use the stronger expression. That is to say, she implicates that the stronger expression does not hold. This is a *Quantity* implicature because appeal is made to the existence of a more informative utterance which the speaker chose not to use. It is a *generalised* implicature inasmuch as one can draw up scales of these expressions where use of weak items in the scale implicates the negation of stronger elements ([Hirschberg

[4] Remember that, as pragmatic inferences, these are defeasible.

[5] Members of scales are listed in ascending strength from left to right.

[6] This also has a (scalar) implicature based on the scale <'or', 'and'>, i.e. 'Sir Andrew is not both a knight and a knave'.

1985] defines scales more fully). Again, (8b) and (9b) list generalised Quantity implicatures of (8a) and (9a) but these cases are referred to as *clausal implicatures*. They arise in a similar way to scalar implicatures but here the speaker implicates simply that she is unable to make the stronger utterance (and not that the stronger utterance is actually false): "These will arise when a compound sentence p has a constituent sentence q such that p entails neither q nor *not*-q and, on Gazdar's theory, presupposes neither as well. As there is usually a similar assertion that would entail q, or its negation, the speaker is presumed not to know whether q is true or whether q is false." [Atlas & Levinson 1981, p.39]. Both scalar and clausal implicatures make "what is communicated" *more definite* than what is said [*ibid.*, p.35].

1.4.4 Conventional Implicature

Conventional implicatures are also non-truth-conditional aspects of meaning but they do not arise in connection with any conversational principles: they are conventional in the sense that they are triggered by the use of certain words or forms of words[7]. The senses of the following words trigger conventional implicatures:

- 'therefore', 'but' [Grice 1975], 'even' [Kempson 1975], 'yet' [Wilson 1975]
- Discourse-deictic items[8]: 'however', 'moreover', 'besides', 'anyway', 'well', 'still', 'furthermore', 'OK', 'so' [Levinson 1983]
- Social-deictic items: 'sir', 'madam', 'mate', 'your honour', 'sonny', 'hey', 'oi', the 'tu'/'vous' distinction in French and honorifics in other languages [Levinson 1983][9]

For example ((a) is the utterance, (b) is the truth-conditional meaning and (c) is the conventional implicature):

(10) a. "*Helena loves Bertram but Bertram does not love Helena.*"
b. (Helena loves Bertram) $\wedge$ (Bertram does not love Helena)
c. 'There is a contrast between the two facts that Helena loves Bertram and Bertram does not love Helena.'

1.4.5 Distinguishing Types of Inference

The inferences I have just described cannot be distinguished on the basis of their form. But they are generally accepted as having different properties as set out in the table below:

[7] Richard Ogden has suggested to me that intonation can convey conventional implicatures too.

[8] These are words that in AI are sometimes called clue-words [Reichman-Adar 1984].

[9] That certain words are vulgar or slang is also carried as a conventional implicature.

	Entailment	Presupposition	Conventional implicature	Conversational implicature
Conventional	no	yes	yes	no
Calculable	no	no	no	yes
Cancellable in positive utterances	no	no	no	yes
Behaviour in negative utterances	may not survive	survive but are cancellable	survive but are not cancellable	may change and are cancellable
Detachable	no	yes	yes	no
Reinforceable[10]	no	asymmetrically	no	yes

I will go through these properties in turn (see [Sadock 1978] for critical assessment of these properties: I do not describe their problems here).

- **Conventional**

Conversational implicatures and entailments are *non-conventional.* By contrast, presuppositions and conventional implicatures, being triggered by the actual form of the utterance, are conventional. This means that presuppositions and conventional implicatures are triggered by the use of certain words or forms of words, whereas entailments and conversational implicatures can only be computed with reference to utterance meaning.

- **Calculable**

Conversational implicatures are *calculable.* This is to say that conversational implicatures can be 'worked out', i.e. if utterance U conversationally implicates proposition P, then it should be possible to give an argument as to why P arises from U. For example ((a) is the utterance, (b) is the conversational implicature and (c) is the argument which justifies the implicature):

(11) a. (In response to the question "*Where's Sir Toby?*"): "*Well, the pubs are open.*"
b. 'Sir Toby is in the pub.'
c. In response to the question "*Where's Sir Toby?*", (11a) seemingly violates the maxims of Quantity and Relation. But, assuming the speaker *is* being coöperative, then she could be intending to convey (11b) which restores her utterance to being a coöperative one.[11]

On the other hand, other inferences are not like this. There is no argument you can give to explain why U conventionally implicates, entails or presupposes some P. For example ((a) is the utterance, (b) is a conventional implicature):

(12) a. "*Helena loves Bertram but Bertram does not love Helena.*"
b. 'There is a contrast between the two facts that Helena loves Bertram and Bertram does not love Helena.'
c. There is no argument one can give to justify (12b) — the conventional implicature is triggered by use of the word "but".

[10] Assuming no heavy stress in the reinforcing phrase.

[11] This is a rather informal presentation of the argument.

- **Cancellable in positive utterances**

Conversational implicatures, unlike all other inferences, are *cancellable* in positive utterances. For example, the conversational implicature that was implicated in (7), i.e. that Macbeth met no more than three witches from the utterance "*Macbeth met three witches*", can be explicitly cancelled as follows:

(13) "*Macbeth met three witches and maybe more.*"

But conventional implicatures cannot be cancelled without causing some infelicity:

(14) ??"*Helena loves Bertram but Bertram does not love Helena and, of course, these facts are not incompatible.*"

Nor can we cancel entailments or presuppositions in positive sentences without contradiction:

(15) a. *"*The fool has five teeth missing but he doesn't have a tooth missing.*"
b. *"*The Prince of Denmark is mad but there isn't a Prince of Denmark.*"

- **Behaviour in negative utterances**

In negative utterances entailments generally do not survive (16a), presuppositions generally do survive (16b) but may be explicitly cancelled (16c) and conventional implicatures survive (16d). Conversational implicatures may not be the same as they would in a positive utterance. For example, (16e) conversationally implicates (16f) (compare this to (7)):

(16) a. "*The fool doesn't have five teeth missing.*"
b. "*Lear doesn't regret he hit the fool.*"
c. "*Lear doesn't regret he hit the fool because he didn't hit the fool.*"
d. "*It isn't true that [Helena loves Bertram but Bertram doesn't love Helena].*"
e. "*Macbeth didn't meet three witches.*"
f. 'Macbeth met at least one witch.'

- **Detachable**

Conversational implicatures and entailments are *non-detachable.* Suppose U_1 and U_2 are different but synonymous utterances. Non-detachability of a conversational implicature or entailment means that if U_1 expresses P_i which conversationally implicates or entails P_j, then other ways of expressing P_i, such as U_2, will also implicate or entail P_j. Since conventional implicatures and presuppositions are triggered by the use of certain words of phrases, they are detachable: the use of a synonymous expression might lose the inference.

- **Reinforceable**[12]

Conversational implicatures are *reinforceable.* A conversational implicature can be made explicit without a sense of redundancy. For example, (6a) and its implicature (6b) can be conjoined together in either order without causing an infelicity:

(17) a. "*Some, but not all, of the fairies like Oberon.*"
b. "*Not all of the fairies like Oberon, but some do.*"

Conventional implicatures and entailments may not be reinforced without a sense of redundancy ((18) and (19) respectively):

(18) ??"*Helena loves Bertram but Bertram doesn't love Helena and there is a contrast between these two facts.*"

(19) ??"*The fool has five teeth missing and he has one tooth missing.*"

There is an asymmetry in reinforcement of presuppositions:

(20) a. ??"*Lear stopped hitting the fool and he had been hitting him.*"
b. "*Lear had been hitting the fool and he stopped hitting him.*"

Levinson says [1983, footnote 17, p.120] that presuppositions and entailments (and I would include conventional implicatures) might in fact be freely reinforceable provided the reinforcing phrase has heavy stress:

(21) "*Lear stopped hitting the fool and he had been HITTING him.*"

But as Levinson notes, the stress might produce an additional implicature that Lear had been hitting the fool quite vigorously. If this is so, then the second clause *does* convey extra information; this would explain why we do not find (21) anomalous.

This concludes the review of inference types. To conclude this chapter I give overviews of the system described in this thesis and the problems of presuppositions that must be accounted for.

1.5 Overview of T42

This section gives an overview of T42, the system derived from [Haddock 1987a, 1987b] used in this thesis as a base for presupposition processing. The description is deliberately very schematic. Individual modules are explained in detail in Chapter 3 and a full example is given in Chapter 4.

Figure 1.1 shows T42's structure and should be viewed in conjunction with the rest of this section. The processing cycle begins with an utterance being input to the system. The utterance is processed one word at a time in

[12] This property was introduced by Sadock [1978] while the rest were introduced by Grice [1975].

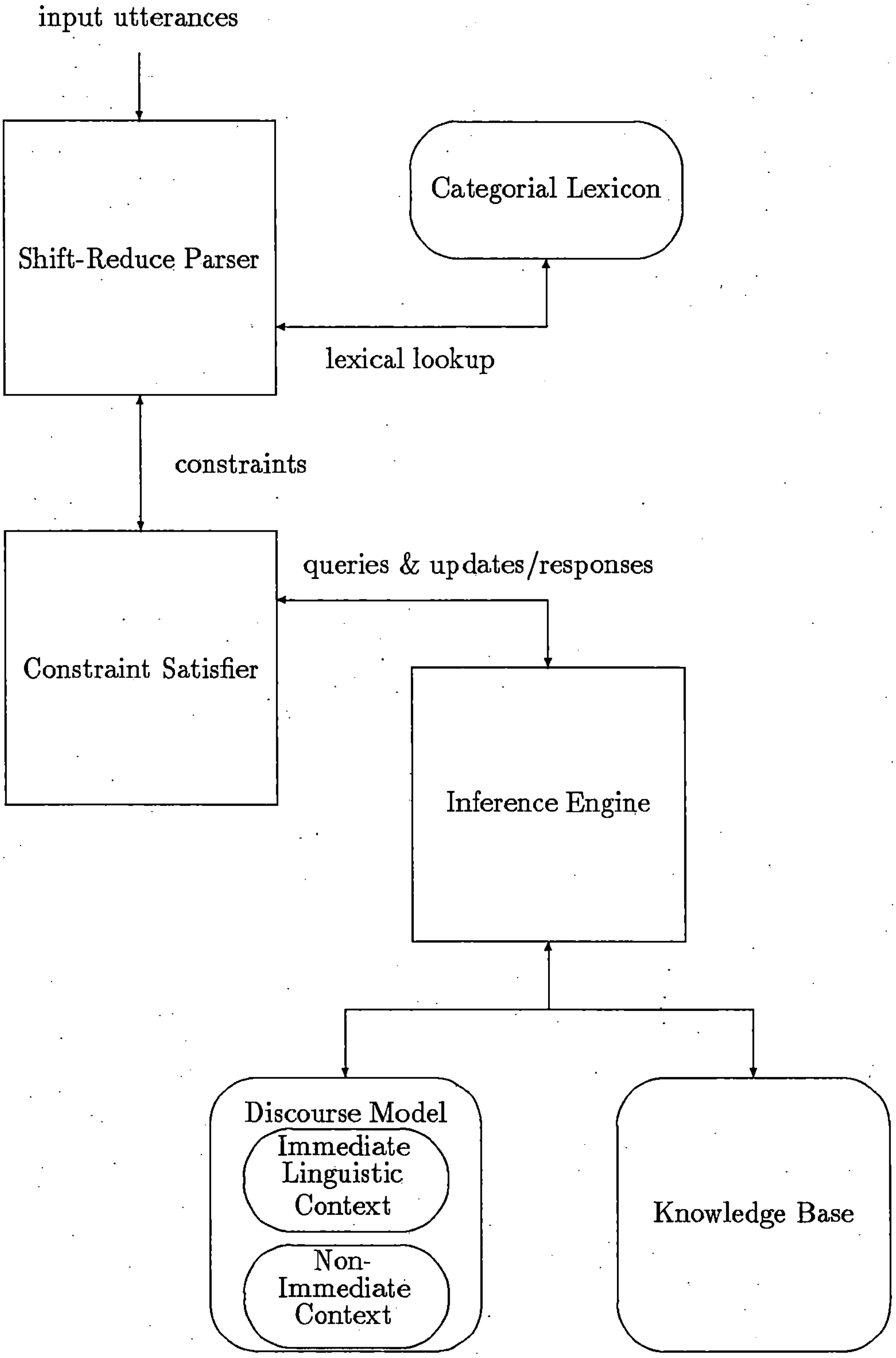


Figure 1.1: System Structure

strict left-to-right order through the utterance. A word is 'fully' processed before processing of the next word may begin. A word is fed to the shift-reduce parser, which retrieves category and lexical sense information from the lexicon. The lexical sense information comes in the form of semantic constraints which are fed to the constraint satisfier. The constraint satisfier has to determine whether each constraint is consistent with those it has already been fed for this analysis. This often involves using knowledge from immediate linguistic context, non-immediate context and the knowledge base. If the constraint satisfier is 'happy', it allows the parser to proceed with the analysis. If it is 'unhappy', it tells the parser to discard the current analysis and try an alternative one.

Assuming the constraint satisfier is 'happy', the parser is allowed to read in and process the next word. Before going onto a third word though, the parser will attempt to combine the first two words into a syntactic constituent. If this cannot be done, this parse must be discarded and an alternative tried. If it can be done, it is likely that this will generate additional semantic constraints and these are immediately fed to the constraint satisfier. If the constraint satisfier is still 'happy', the third word will be processed. Once the semantic constraints have been accepted by the constraint satisfier, and before going onto the fourth word, the parser will try to combine the third word with the phrasal constituent that it has built up for the first two words. Processing continues in this way until all the input is used up: the parser only reads in another word when the current word has been fully processed.[13]

An analysis is successful if the parser recognises the input utterance as a well-formed sentence and if the constraint satisfier deems the analysis acceptable. In this case, the constraint satisfier sends some updates from the immediate linguistic context to the non-immediate context. These updates will represent the utterance reading.

Certain of the constraints sent from the parser to the constraint satisfier will indicate that a presupposition trigger has been encountered. This tells the constraint satisfier that these constraints must be treated as a Constraint Satisfaction Problem that must be solved using entities from either immediate linguistic context or non-immediate context, collectively referred to as the discourse model. If this problem cannot be solved, the presupposition fails. This is the sense in which presuppositions are preconditions.

- **Implementation Details**

Full details of T42's design are given in subsequent chapters of this thesis. The design is implemented in Common Lisp and has been run on a variety of hardware. The programs occupy about 98 Kbytes when compiled (representing about 3000 lines of source code), excluding the categorial lexicon and the knowledge base.

T42 has only been tested on simple examples, most of which are given in the body of this thesis. It takes a sentence from the user, computes one or

[13] By "fully processed" I mean only that all modules of T42, as presently defined, have done as much as they can.

more readings, and then offers the user a choice about which, if any, of the readings should be retained, before prompting the user for the next utterance. For the very simple examples given in the thesis, the processing cycle takes up to about 60 seconds. For larger examples, performance worsens dramatically.

1.6 Overview of Presupposition Issues

"Presupposition" has had a long and confusing history. It can be traced through the work of a number of philosophers (e.g. [Frege 1892 (1975)], [Russell 1905 (1975)], [Strawson 1950]), linguists (e.g. [Kempson 1975], [Wilson 1975], [Gazdar 1979]) and computational linguists (e.g. [Weischedel 1979], [Mercer 1987]). Some writers such as Kempson and Wilson have claimed that a special account of "presupposition" is not necessary: presuppositional phenomena are instances of other phenomena (e.g. entailments of conversationally implicated readings). This thesis argues that presuppositions cannot be 'explained away' in terms of other phenomena. But it also argues that, contrary to the way they have hitherto been viewed in natural language processing, presuppositions are not inferences *tout court*, but are rather inferences that must be used as *preconditions*. I also show that this approach is easily accommodated by incremental NLP systems such as T42. Indeed, T42's processing of definite NPs, to be described in earlier chapters of the thesis, will be revealed as an exemplary form of presupposition processing.

Archetypically, (22c) is said to be a presupposition of (22a) and of (22b):

(22) a. The Prince of Denmark smiled.[14]
b. The Prince of Denmark didn't smile.
c. 'There is a Prince of Denmark.'

Some would say that (22c) is a presupposition of (22a) and of *one of the readings of* (22b); others would say that (22c) is simply an entailment of (22a) and of one of the readings of (22b); others still would say that (22c) is an entailment of (22a) and a presupposition only of (22b).

I will be claiming that (22c) (or something like it) *is* a presupposition of (22a) and of (22b) (which I will take to be semantically unambiguous). In most theories, if the presupposition is satisfied, then (22c) will also be entailed in both cases. However, if (22a) or (22b) are embedded into complex sentences, these entailments, and on certain theories these presuppositions also, may be lost. For presentation purposes I shall normally call (22c) a presupposition even when describing theories which might themselves have rejected this label.

Concentrating on this simple example to begin with, I will state the problems that I believe have arisen in trying to characterise the relationship between (22a), (22b) and (22c).

[14] Pending further discussion in later chapters, I have deliberately refrained from using my conventions to indicate whether (22a) and (22b) are sentences, utterances or propositions. I am, however, taking (22c) to be a proposition.

1. What is it that has presuppositions? Some would say that it is speakers that presuppose propositions such as (22c). Others would define presupposition as a relationship between utterances and propositions, between sentences and propositions, or between propositions and propositions. In the last of these cases, we might ask whether it is propositions characterising utterance readings or propositions characterising sentence readings that presuppose things.

2. Given that (22c) is a proposition, is it to be regarded as an assumption of the speaker of (22a) and (22b), or is it an inference that can be drawn from (22a) and (22b), and if it is, is it the kind of inference that should be treated as a precondition of (22a) and (22b)?

3. Should presuppositions be defined from the speaker's point of view or from the hearer's point of view? This question is not independent of the previous one. If presuppositions are inferences (including those that are preconditions) then we are taking the hearer's point of view, i.e. they are inferences the hearer draws from an utterance reading, and perhaps more specifically preconditions the hearer must be able to satisfy in order to get an utterance reading. If presuppositions are assumptions then we are taking the speaker's point of view.

4. What happens if the presupposed proposition is not satisfied, and is this the same as the presupposition being false? If (22c) is an inference and is false, what does this tell us about (22a) and (22b); if (22c) is a precondition and it fails, what status do (22a) and (22b) have; if (22c) is an assumption, does it matter if it is a false assumption?

5. How is (22c) 'adduced' from (22a) and (22b)? Most theories suggest that certain elements of the semantic representation of the sentence or utterance "trigger" the production of presuppositions.

6. How do we explain the 'background feel'[15] of (22c)? In theories where presuppositions are assumptions or preconditions this is not a problem: they have a 'background feel' almost definitionally. In theories where presuppositions are inferences but not preconditions their 'uncontroversial status' must be explained.

7. While both (22a) and (22b) presuppose (22c) (it perhaps being a definitional property that both the positive and negative forms of an utterance share the same presuppositions), there is an asymmetry to explain. It is often alleged that utterances of negative sentences do not always 'retain' their presuppositions. Utterances of simple positive sentences always do. For example, (23a) is infelicitous but (23b) is not: (23b)'s presupposition can be 'removed' ("cancelled", "blocked" or whatever) where (23a)'s cannot. However, these "cancellation" cases, where a

[15] "...let us note that there is a way in which there is an intuitive unity to this set of inferences. For the basic intuition is that they are all in some important sense *background assumptions* against which the main import of the utterance ... is to be assessed." [Levinson 1983, p.180]

negative utterance does not have the same presuppositions as the corresponding positive utterance are 'dispreferred' uses of negative utterances.

(23) a. *The Prince of Denmark smiled — but there isn't a Prince of Denmark.
b. The Prince of Denmark didn't smile — because there isn't a Prince of Denmark.

Both the asymmetry between (23a) and (23b) and the "marked" or "dispreferred" status of (23b) require explanations.

8. What are the presuppositions of complex sentences (i.e. conjunctions, disjunctions, conditionals, etc.)? They are not necessarily straightforwardly the union of the presuppositions of their constituents. For example, (24) probably does not presuppose (22c):

(24) If there is a Prince of Denmark, the Prince of Denmark smiled.

These are complicated questions. But before proceeding I shall simply say here, without elaboration, how the formulation of presupposition I develop in this thesis answers each of the questions. In T42:

1. *Utterance readings* have presuppositions.
2. Presupposed propositions are *preconditions* on the hearer understanding an utterance reading.
3. Presuppositions are defined from the *hearer's* point of view.
4. If a presupposed precondition is not satisfied, *the reading is ruled out.* This does not rule out the possibility of there being other readings whose presuppositions are satisfied. If all readings are ruled out, the utterance might be infelicitous (see Chapter 7 for discussion). But importantly, presupposition failure is not necessarily the same as a presupposition being false in, e.g., the 'real' world.
5. The lexical semantics of words may contain presupposition *triggers.* The presupposition is triggered in *all* circumstances where the word is used.
6. Presuppositions have 'uncontroversial status' because they are *preconditions.*
7. Utterance of a positive sentence or its negative counterpart will *always* trigger any presuppositions. Hence positive utterances and negative utterances *share* the same presuppositions. Presupposition "cancellation" in negative utterances is metalinguistic: it indicates something about the model-theoretic interpretation that should be given to statements in the discourse model. A full explanation of this is given in Chapter 8.

8. Presuppositions are *always* triggered in complex sentences. There is no notion of cancellation. These presuppositions, as preconditions, must be satisfied if a reading is to 'come off'. However, complex sentences do not always inherit the *entailments* of their constituent clauses. A full explanation of this is given in Chapter 9.

::::

In this chapter I have introduced the problems of presuppositions, and have suggested that these can be solved using a notion of presuppositions as preconditional inferences. I have also suggested that on this conception presuppositions are best computed by an incremental language processing system. The next chapter looks further at what I mean by an 'incremental' system.

Chapter 2

Incremental Processing: Interleaving Syntactic, Semantic and Pragmatic Operations

In this chapter, I explain what I mean by *incremental* processing and argue in favour of its use. I begin by briefly reviewing 'traditional' system designs and compare these to incremental systems from both a linguistic point of view and a software engineering point of view. I follow this by considering the arguments for and against incremental systems. As part of this I consider cases of "pragmatic intrusion" [Levinson 1988] which, I believe, offer a new argument from linguistics that not only further confirms the advantages of incremental systems but actually *demands* their use. Finally, I briefly review two other incremental systems and indicate the linguistic problems which incremental processing is supposed to help them to solve.

2.1 Types of NLP Systems

Classifying natural language processing (NLP) systems is difficult. Even though I am only concerned with systems that do language analysis not language generation, there is still a large number of possible axes of comparison (e.g. whether different types of knowledge are kept separate or not; whether different types of knowledge are represented uniformly or not; whether knowledge is represented procedurally or declaratively; whether processing is done by separate modules each responsible for accessing different types of knowledge or whether a single module accesses different types of knowledge in a relatively undifferentiated way; how and when the control structure invokes different modules; etc.). Points on different axes might often or even necessarily cooccur (e.g. if knowledge is represented uniformly then processing is likely to be done by a single, undifferentiated process). It can also be difficult to determine where on a particular axis a system lies (for example, it may not be possible to say whether different types of knowledge are, in fact, represented 'separately'). Since my main aim is merely to explain what I mean by 'incremental' I have chosen not to give a detailed classification using all these axes of comparison. Instead, I classify systems according to the degree of interaction between syntax, semantics and pragmatics which they exhibit.

Ritchie [1983] similarly divides systems using a classification concerning the interaction of syntax and semantics, and distinguishes three classes: homogeneous, sentence final and interleaved. I have found this a useful classification and so I have adopted it. But I have also chosen to subdivide Ritchie's third class, the interleaved systems, into three, according to the degree of interaction used. Here is the result:

1. **Homogeneous:** Uniform representation of different types of knowledge, undifferentiated processing;

2. **Sentence Final:** Distinguishable representation of different types of knowledge, distinguishable processes with sequential execution;

3. **Interleaved:** Distinguishable representation of different types of knowledge, distinguishable processes with interwoven execution. I distinguish three sub-cases:

   (i) No feedback: although processing is interwoven, there is no feedback from one process to another;
   (ii) Limited feedback: interwoven processing where simple semantic checks can rule out certain parses;
   (iii) Semantic and pragmatic feedback: interwoven processing where the results of semantic checks, consultation of context and inferencing can be fed back to help fix, choose or augment the full utterance reading. *These are the systems for which I reserve the term 'incremental'.*

I will now review these types of system in more detail, stating some of their advantages and disadvantages. I believe that the disadvantages that I state for a class apply to systems of that class in general and are not merely problems associated with particular implementations.

2.1.1 Homogeneous Systems

Homogeneous systems are best exemplified by those that emerged from the work on Conceptual Dependency at Yale University (see, e.g., [Rieger 1976], [Schank 1975], [Schank 1980], [Schank & Rieger 1974]) and by Small's "Word Expert Parser" [Small 1983]. In these systems there is no clear distinction between syntactic, semantic and pragmatic knowledge (which are all usually represented procedurally) and no clear distinction between syntactic, semantic and pragmatic processing. A meaning structure is built directly from the input string in a single pass over that string.

Speaking roughly, such a system works through the input from left-to-right, maintaining a global data structure, such as a Conceptual Dependency network, which records the meaning representation as it is built. Each word in the lexicon has associated with it some procedural knowledge reflecting the word's semantic contribution. As each word is read in, its lexical procedure is retrieved and executed. The procedure might consist of a set of test-and-action pairs: an action to update the global data structure will only be performed if its test is or becomes true. For ambiguous words the meaning structure should cue one sense rather than another and hence fire one action rather than another. This scheme is susceptible to being miscued [Ritchie 1983]; these systems typically have no recovery mechanism from this.

Authors of these systems sometimes claim that the systems do *no* syntactic processing. If this were so, the systems would not be able to distinguish "Brutus killed Caesar" from "Caesar killed Brutus". It is probably not so. As Ritchie says "...it is rather misleading to term such systems 'non-syntactic' or 'wholly semantic', since they actually use a mixture of syntactic and semantic constructs. Their main distinguishing feature is the lack of a formal separation between these constructs — all grammatical information is treated as being qualitatively the same." [Ritchie 1983, p.201]. Boguraev [1979] points out that, with such a weak, implicit notion of syntax, these systems are unable to give different interpretations to examples along the lines of "Brutus stopped to help Caesar" and "Brutus stopped helping Caesar". Furthermore, there is no indication of how such systems might detect and resolve certain structural ambiguities such as those concerning prepositional phrase attachment. These systems' notions of syntax are so weak and their processing is so 'verb-centred' that these structural ambiguities might be undetectable.

In addition to these linguistic problems, there are problems of engineering such systems. The construction of Word Experts, for example, was an impossibly difficult task. The Expert for "throw" covered 6 pages and was still felt to be inadequate. Whether criticisms of the engineering of these systems strike home is an open question. The proponents of these systems take the position that their systems have some sort of psycholinguistic plausibility. Maybe such systems do not have to satisfy principles of good design. I do not agree. Engineering problems can lead to linguistic problems: the systems capture few linguistic generalisations and so are not easily extensible. It is perhaps not surprising that work that has followed on from the work at Yale has relied on introducing a separate syntactic processing phase (see [Cater 1981] in which Schank's group's work is extended). However, I should say that my main criticisms here ignore the main contributions of this work on homogeneous systems, namely the investigations they allowed into the issues of lexical semantics.

2.1.2 Sentence Final Systems

In the systems Ritchie terms *sentence final*, a first pass over the input string builds one or more full syntactic structures for the input string, and a second process converts these to semantic representations which are then submitted for pragmatic processing. Thus these systems have distinct knowledge sources and distinct processing stages. One stage must complete the structures it is building before these structures are passed to the next stage. Different processes are not interwoven and there is thus no possibility of feedback to syntactic processing from semantic or pragmatic processing. An example system is SRI's portable natural language database front end, TEAM [Martin et al. 1983].

Systems of this kind can take one of two approaches to problems such as structural ambiguity. The simplest solution, taken by TEAM, is to produce multiple parses, which are mapped to multiple semantic representations. Subsequent modules may then use syntactic preferences, lexical preferences, contextual information and/or world knowledge to rule some of these out. In this approach *all* parses must be produced even though only one will be chosen as the intended utterance reading: as there is no feedback, these systems are forced to compute unneeded structures. The alternative solution to problems of ambiguity, which avoids computing these unwanted structures, is to compute, where possible, a single 'vague' structure which, in some sense, is general enough to capture all the readings of the sentence: it should be possible to 'pull out' each reading from this vague structure.

Sentence final designs on the whole have the advantage that they have simple control structures, and because of this they have been extensively applied in practical contexts. Their disadvantage is that either they compute unneeded structures or we have to face the difficulties of devising vague structures to encapsulate ambiguities. One of the tenets of this thesis is that there are linguistic phenomena, presupposition in particular, which can be accounted for more naturally by systems other than the sentence final ones.

2.1.3 Interleaved Systems

In *interleaved* systems there is, ideally, only one pass over the sentence. During this pass, the syntactic processor sends *partial* syntactic structures to the semantic analyser for processing. The semantic analyser may also request pragmatic processing. When semantics and pragmatics are done with this partial structure, syntactic analysis may proceed. This interleaving has become more common as systems which use unification-based grammar formalisms and unification-based processing have become more popular. Such systems represent syntactic and semantic knowledge uniformly and use the same process of unification to build both syntactic analyses and semantic representations for an utterance in tandem. Despite the uniformity of knowledge representation and processing mechanism such systems do not fall into the homogeneous class: syntactic and semantic knowledge are still distinguishable.

The significance of this interwoven processing really depends on the extent to which syntactic, semantic and pragmatic analysis provide *feedback* to one another to focus future processing.[1] I have therefore split these systems into three classes characterising the degree of feedback:

- **No feedback**

In the simplest case, semantic and pragmatic processing have no bearing

[1] Comments made here about interleaving processes apply equally to parallel processing.

on syntactic processing or on each other. Nothing semantics discovers, for example, would be used to influence the parser. Such systems are little more than sentence final systems with more complicated control structures.

- **Limited feedback**

More common are systems where failure of simple semantic checks (e.g. selection restrictions using semantic features) is used to rule out certain parses. There are many systems of this kind. Unification-based systems often incorporate such checks but the Cascaded ATN systems probably fall into this class too, e.g. [Bobrow & Webber 1980], [Boguraev 1979], [Ritchie 1980] and [Woods 1980].

- **Semantic and pragmatic feedback ("incremental")**

I reserve the term "incremental" for systems in this subclass. For a system to qualify for inclusion in this class there must be more feedback than the results of simple semantic checks as above. For example, semantics might also check for self-consistency; pragmatics might help to rule out parses on the basis of lack of plausibility, failure of presuppositions or lack of consistency with context, and might also find referents for referring expressions or augment a logical form with pragmatic inferences. T42 falls into this class of system, as do the two systems reviewed at the end of this chapter. Obviously there is not necessarily a principled or clear dividing line between these incremental systems and those I have classified as having 'limited feedback'. However, a division of these sorts does seem to have arisen in terms of the systems reported in the literature.

In the next section I argue the case for incremental systems.

2.2 The Case for Incremental Systems

Two main advantages ((1) and (2) below) have been adduced for incremental systems (see, e.g., [Ritchie 1983]) and I wish to add a third advantage ((3) below):

1. Incremental processing may be a more *psychologically plausible* model of human language understanding;

2. Incremental processing may *save computation* by avoiding unnecessary structure-building;

3. Incremental processing may give a *simpler account of certain linguistic phenomena.*

I will look at these alleged advantages in turn.

2.2.1 Psychological Plausibility

Some might claim that incremental systems are psychologically plausible models of human language processing. While I can agree that a system that works from left-to-right through its input, with interleaved syntactic, semantic and pragmatic processing, might be a *more plausible* model than, say, a sentence final system, this does not qualify it, in my opinion, to being described as 'plausible' *tout court*. For example, I have no evidence that T42's *degree* of interleaving (i.e. after each word) is plausible. It may be that interleaving at clause and noun phrase boundaries would be more appropriate. Similarly, T42 does not exploit 'expectations' at all in its processing and this is, at least intuitively, implausible. I treat claims of plausibility with the utmost scepticism and choose not to base my case for incremental systems on such claims. This is not to say that psycholinguistic results, when available, should not be incorporated into NLP systems. Indeed, Chapter 4 indicates that Haddock's system, on which T42 is based, made use of such a result.

2.2.2 Saving Computation

I indicated that systems that do not have feedback from semantic and pragmatic modules to the parser may have to compute all structures for an ambiguous sentence before ruling some of them out (unless they compute a single 'vague' structure), whereas systems with feedback might reduce the number of unnecessary structures that need to be built. Ritchie casts doubt on this supposed advantage of incremental systems.

For example, consider the argument which says that selection restrictions can save computation. In processing a sentence that begins "The landscape painted ...", since landscapes do not paint, the early application of this semantic knowledge (selection restriction) can tell us that this should not be parsed as 'Subject + Main-Verb' but as 'Subject + Passive-Modifier' (as in "The landscape painted by Orsino fetched a good price."). But Ritchie raises two problems. First, since such restrictions and checks are often very weak and may be violated, there may not be many cases where this can help. Secondly, "[a] phrase which seems to violate this simple form of semantic consistency when considered in isolation, may function perfectly smoothly in context." [Ritchie 1983, p.208]. For example (from [Ritchie 1983]):

(1) a. "It is nonsense to speak of a rock having diabetes."
b. "He says that he poured his mother into an inkwell."
c. "A round square is a puzzling idea."

Example (1c) shows that trying to patch this up (e.g. switching off checking when an embedding construction is spotted), will not work: in this case the information which tells us to switch off the checking appears *after* the questionable phrase. These and similar examples have led to approaches based on preferences rather than constraints (e.g. [Wilks 1975, 1978]). Ritchie claims that these problems with selection restrictions are but one case where interleaving has questionable value. He goes on to question the usefulness of other forms of interleaving, e.g. allowing general inference during parsing.

It does have to be admitted that it is not obvious that interleaving *will* save structure computation. More than this, there may be cases where interleaving causes more work to be done than would be the case with a sentence final system. The first case of this that comes to mind is where the sentence being processed is syntactically ill-formed: early parts of the sentence might demand heavy use of semantic and pragmatic processing which might be wasted given the subsequent ill-formedness, e.g.:

(2) *"Portia, who owned the portrait in the casket, loving Bassanio."

Here, effort expended in consulting the context to process the relative clause and prepositional phrase is wasted since the main clause verb phrase is ill-formed. This is not too worrying: first, such cases are pathological anyway, and secondly the earlier processing will not be wasted if our systems, like humans, go on to attempt to make sense of the ill-formed utterance.

More worrying are cases where, while a structural ambiguity *can* be resolved with reference to semantics and context, it might also be resolvable by syntax alone. Given that using semantics and context is computationally expensive, syntax alone would be the cheaper solution, e.g.:

(3) "The merchant sold the wheat for a lot of money wanted to kill Launcelot."

Here, use of context might resolve the local ambiguity arising from trying to parse "sold" as a tensed verb when it should be parsed as a past participle, but syntax alone can sort this out more cheaply[2]. However as Marcus points out: "There's a nice and extremely useful property of people, that when they're confused they know they're confused very immediately, and they let you know." [Pulman 1987, p.205]. So, Marcus says, in practical systems you may need to do things as soon as possible: in a speech recognition system, you should not let the user go on for 30 seconds before saying "Sorry I didn't understand the second word of the sentence."

2.2.3 Accounting for Linguistic Phenomena

The final, and by far the best, reason why incremental systems should be preferred over other designs is that they give a natural and simple account of certain linguistic phenomena. [Ogden 1989] and [Hirst 1987] go some way towards showing this for lexical ambiguity. [Mellish 1985], [Haddock 1987a, 1987b] and [Hirst 1987] have all made a strong case for incremental processing to help make attachment decisions (see Chapter 4) and this thesis shows that these ideas can be extended to presupposition processing. To show that this is not an isolated call for such systems, I shall briefly review, in the next section, some new work in linguistics that also would seem to demand the interleaving of semantic and pragmatic processing. The work is by Levinson [1988] and it suggests that the process of drawing *implicatures* should be interleaved with other forms of processing. I have not implemented anything along these lines: my work has concerned presuppositions not implicatures. I include Levinson's ideas here to show that there are other reasons why incremental processing might be useful.

[2] This was pointed out by Ron Kaplan in [Pulman 1987].

2.3 Pragmatic Intrusion

It is not my intention to try to defend Levinson's notion of pragmatic intrusion. It is a new theory and still under development and so may yet be further modified a great deal. Rather, I include it here to show that my belief that presuppositions need to be computed incrementally is complemented by work that suggests that implicatures should be computed incrementally too. In Chapter 1 I briefly reviewed the notion of conversational implicature. Implicatures, I said, were introduced by Grice to account for certain of the ways in which utterances can convey more than their straightforward truth-conditional content. They are thus part of non-truth-conditional meaning. Conventionally, truth-conditional meaning has been viewed as the input to the pragmatic processes that produce implicatures. Levinson [1988] claims that there is a problem with this viewpoint, which is that some implicatures must be computed *as part of* the process of producing the truth-conditional meaning: implicatures can intrude into the logical form. I will construe Levinson's argument as giving further support to the case for interleaving semantic and pragmatic processing. Levinson concentrates on generalised conversational implicatures in his exposition, but he recognises that particularised conversational implicatures may do all the things he shows generalised ones may do. He chooses not to deal with particularised ones explicitly as he feels that generalised implicatures are more interesting from a linguistic point of view.

[Levinson 1988]'s sub-title is "How you can't do semantics without first doing pragmatics". He criticises what he calls the "received view" of the role of pragmatics in linguistic theories where it is claimed that semantics is done without first doing any pragmatics. The "received view", following Grice, is that semantics tells us 'what is said', the truth-conditional content of the utterance. (In fact, Grice assumes that referents have been resolved, indexicals have been fixed, disambiguation has been done, ellipses have been unpacked, etc.). 'What is said' is the input to pragmatic reasoning, which produces 'what is implicated' and thus the full utterance meaning is obtained. I am taking the "received view" to be analogous to the sentence final approach.

But, Levinson says, pragmatic knowledge must in fact be used earlier than the received view suggests, for two main reasons:

1. Resolving referents, fixing indexicals, doing disambiguation and unpacking ellipses all involve pragmatic reasoning anyway;

2. There are cases where pragmatics must intrude into the truth-conditions. If this is not allowed to happen, bizarre and counter-intuitive truth-conditions will result.

What Levinson is saying is that in certain cases the "received view" will not generate an adequate logical form. The form it generates will not adequately capture the truth-conditions of the utterance; it will be so weak that the form cannot be adequately assessed for truth in a world. He is saying that some pragmatics must be done in order to get an adequate set of truth-conditions: in general, pragmatic inference is crucial to semantic interpretation[3].

I will briefly cover the main examples which Levinson uses to demonstrate pragmatic intrusion: comparatives, conditionals, metalinguistic negation and reference resolution.

- **Comparatives**

In Chapter 1 I gave an example (number (5) in that chapter) in which "and" was implicated to mean 'and then'. If this implicature is not allowed to intrude into the logical form of the following example, bizarre truth-conditions result, contrary to intuitions:

(4) "*Driving home and drinking three beers is better than drinking three beers and driving home.*"

"Sentences of the form 'A is better than B' will be anomalous, indeed contradictory (necessarily false), unless the proposition expressed by A is distinct from that expressed by B. But if A has the form 'p and q' and B has the form 'q and p', and *and* is equivalent to the truth functional connective $\wedge$ (and thus $p \wedge q = q \wedge p$), then 'A is better than B' will be necessarily false. But the sentence above has this form; and it is, let us agree, just plain true!" [*ibid.*, p.23]. The reason that (4) is not contradictory, Levinson claims, is that the implicature intrudes into the logical form to give:

(5) 'Driving home *and then* drinking three beers is better than drinking three beers *and then* driving home.'

[Levinson 1988] gives many more examples involving this and other implicatures intruding into comparatives.

- **Conditionals**

Conditionals do not give such clear-cut examples: the problem is deciding what to use as a logical form for conditionals. Using a simple material implication, the truth-conditional content of (6a) is (6b) and yet since one feels inclined to assent to (6a) it cannot mean (6b) to which one would not assent. If the implicature from 'some' to 'not all' is allowed to intrude into and hence strengthen the antecedent a more intuitive logical form results (6c):

(6) a. "*If you ate* some *of the victuals (and no one else ate any), then there must still be some left.*"
b. 'If you ate some or all of the victuals (and no one else ate any), then there must be some left.'
c. 'If you ate some but not all of the victuals (and no one else ate any), then there must be some left.'

[3] Post-semantic pragmatic processing may still be needed to account for the double-barrelled flavour of ironies, metaphor and indirect speech acts.

- **Metalinguistic negation**

Levinson believes that certain cases of metalinguistic negation are better accounted for using a pragmatic intrusion explanation. I will not go into details here as much of Chapter 8 is taken up with issues surrounding negation and I return to metalinguistic negation there.

- **Reference resolution**

The truth-value of a clause cannot be determined without determining the referents of noun phrases. Levinson gives cases where referents cannot be determined without using implicatures. Consider a conversation in which there are two men in the visual fields of the conversational participants, one with three witches near him and one with four witches near him. It is clear that each man has at least three witches near him. If (7) is now uttered,

(7) "*The man with three witches near him is Macbeth; the man with four witches near him is Banquo.*"

any uniqueness condition on 'normal' truth-conditions will not be satisfied: the first NP will have as its 'normal' truth-conditions 'the man with at least three witches near him'. But, *both* men satisfy this description so the uniqueness constraint is violated. If the uniqueness condition on the first NP is to be satisfied, an implicature must be allowed to intrude into the logical form so that the first NP means 'the man with no more than three witches near him'.[4,5]

- **Summary of Pragmatic Intrusion**

My review of [Levinson 1988] does little justice to Levinson's arguments, which he develops at much greater length. Levinson does recognise that with these ideas being in their formative stages they are still open to attack. To some extent the main justification Levinson gives in favour of pragmatic intrusion is the number of types of example he can give that do seem to exhibit the phenomena. Levinson does try to anticipate some of the responses that might be made to his argument and to counter them. I will look at the response that might be most telling.

[4] This does raise the question: what stops an implicature intruding into the second clause also? Levinson is mute on this point.

[5] This is true not simply of *referential* uses of NPs but also *attributive* and *generic* uses (these distinctions are explained in Section 5.2), e.g. the NPs in the following example are used attributively:

"*The man who has two children is prudent; the man who has three is a fool.*"

If implicatures do not intrude, the reading for this will be:

'The man who has at least two children is prudent, the man who has at least three is a fool.'

This means that the man who has three children is both prudent and a fool. To avoid these truth-conditions, implicatures must intrude to give a logical form:

'The man with no more than two children is prudent; the man with at least three is a fool.'

A non-believer in pragmatic intrusion might be able to claim that all the cases can be handled by a straightforward Gricean account anyway with no need for intrusion. For example, utterances of the form "A and B is better than B and A" have truth-conditions which are plainly false. To utter such a false sentence would clearly flout the maxim of Quality and, by the standard Gricean argument, this would trigger an implicature. The implicature would indicate that the utterance conveys something other than the straightforward truth-conditions. Intrusion of pragmatic processing and implicatures into the truth-conditions themselves is thus not necessary.

However, Levinson argues that it is not always obvious that a falsehood will fortuitously arise to trigger a rescuing implicature. For example, Levinson argues that (8a) has truth-conditions (8b) but that an implicature intrudes to give the logical form (8c):

(8) a. "*If it costs £20, I have enough money to get in.*"
b. 'If it costs at least £20, I have enough money to get in.'
c. 'If it costs no more than £20, I have enough money to get in.'

For Levinson the implicature is generated straightaway as a default inference and immediately intrudes into the logical form. (As a default inference it can be retracted if we later find that the implicature was not needed or if later information cancels it). For the "received view", one would have to argue that (8b), the standard truth-conditional content, is false or bizarre enough to violate a maxim and hence trigger an implicature. It is not obvious that (8b) would make this happen. And, even if it could happen, Levinson asks why, on hearing (8a), one does not get a feeling of double-take or repair as one presumably should if normal truth-conditions are being produced, a maxim is being violated and an implicature generated.

Levinson concludes that the "received view" "...is not ...in any way inevitable or intrinsic to the subject matter. It is elegant precisely because it frees us from the need to have a theory of 'control' — an independent theory that specifies how modules may interact. But if pragmatic intrusion is indeed established, this picture cannot easily be maintained (at least for the relation between the modules that do semantic interpretation and pragmatic processing). The obvious proposal is to change the unidirectional monologue between components into a bidirectional dialogue — processing could proceed up to some point, pragmatic processing then be called in the way that a subroutine can be called in a program, and semantic processing can then proceed further. The result: the effects of pragmatic intrusion." [*ibid.*, p.61]. In computational terms this conclusion can be construed as a demand for incremental systems.

2.4 A Review of Incremental Systems

In this section I briefly describe two other systems in which different types of processing are interleaved and in which the effects of one type of processing are fed back to the other types. The systems I review are Pollack &

Pereira's CANDIDE and Hirst's Absity and Semantic Enquiry Desk system. These systems are those I know of which come closest to being incremental in the sense developed in this chapter. CANDIDE interleaves semantic and pragmatic processing; Absity interleaves syntactic, semantic and pragmatic processing. I exclude SHRDLU [Winograd 1972] because its uniform procedural representation of knowledge arguably makes it closer to homogeneous systems than to incremental ones. And I exclude the work of Mellish and Haddock, whose systems are direct precursors to T42: discussion of them is subsumed by the next two chapters.

I have not attempted exhaustive reviews of CANDIDE and Absity; I merely highlight the nature of the systems' incremental processing and how this is used to solve certain problems of natural language processing.

2.4.1 CANDIDE

CANDIDE is a multimodal system for knowledge acquisition [Pollack & Pereira 1988], [Pereira & Pollack 1988], consisting of three modules: a *PATR-II parser*, *CANDIDE-SPI*, which does semantic and pragmatic interpretation to produce logical forms, and *TRANSFORM*, which maps the logical forms into the language of the application system. This is a new system, still under development, so a definitive critique is not possible.

The PATR-II parser produces a "least-commitment parse" for a sentence, i.e. a single vague representation. Syntactic processing is done as a separate initial phase. However, Pollack & Pereira do say: "There are reasons to suspect that ultimately syntactic analysis should be incorporated into the same stage of processing as semantic and pragmatic analysis; in particular, it is difficult to develop syntactically neutral representations for certain constructions such as conjunction." [Pollack & Pereira 1988, footnote 2, p.75].

The SPI module does integrated semantic and pragmatic interpretation. Working on the parse tree and making reference to, and possibly updating, the context, this module is able to produce a logical form for the parse. In doing so it has the potential to resolve prepositional phrase attachment, compound nominal and quantifier scoping problems and to resolve pronominal and definite references. The semantic and pragmatic analysis is driven by the syntax in that SPI works recursively down the parse tree.

Although semantics and pragmatics are integrated, the distinction between compositional semantics and context-dependent pragmatics is maintained. The context-independent semantics are computed by compulsory compositional semantic interpretation rules. However, the semantic rules do not simply specify a piece of logical form. They specify a "conditional interpretation" for a syntactic construction. "... [C]onditional interpretations separate the context-independent aspects of an interpretation from those that are context-dependent. Each conditional interpretation consists of a *sense* and a (possibly empty) set of *assumptions*. As a first approximation, one might think of the sense of a phrase as representing purely semantic information, that is, information that can be adduced solely from the linguistic content of the phrase, no matter in which context the phrase has been uttered. The assumptions then represent constraints relating the phrase's sense to its ultimate interpretation. A *complete* interpretation has an empty assumption set, indicating that all of its dependencies on context have been resolved." [Pollack & Pereira 1988, p.76].

For example, the conditional interpretation of the noun phrase "the dog" might be the pair <x, {bind(x, def, DOG)}>, where the first argument is the sense and the second is the set of assumptions. This expression is read as: the sense is x under the single assumption that x can be bound to an entity of type DOG in accordance with the constraints of definite reference. The conditional interpretation of the verb "bark" might be <BARK(y), {bind(y, arg1, ANIMAL)}> which says that "bark" has sense BARK(y), assuming y can be bound as the first argument of the verb to an ANIMAL. The assumption encodes a selectional restriction. "The dog barks" might have the conditional interpretation <BARK(y), {bind(y, arg1, ANIMAL), bind(x, def, DOG), restrict(arg1, =, x)}>, i.e. the conditional interpretation of "the dog barks" is given by the sense of the verb with an assumption set containing the union of the assumptions of the noun phrase, the verb and an extra assumption, {restrict(arg1, =, x)}, which is a restriction that arg1 of the verb must be equal to the sense of the noun phrase. That is y barks assuming that y is an animal, x is a dog and $x = y$. This example shows the two kinds of assumptions used: "bind assumptions" and "restrict assumptions".

Given conditional interpretations, there are then "pragmatic discharge rules", which change a conditional interpretation by eliminating assumptions with respect to context. They may also change the context. The rule for discharging an assumption of the form bind(*var*, def, Pred), for example, is that if there is a unique contextually occurring entity which satisfies Pred, then the assumption may be removed from the set of assumptions and the sense may be changed to reflect the binding. Thus, for the example, "The dog barks", if we uniquely find the individual Crab of type DOG[6] we can discharge the second bind assumption to give <BARK(y), {bind(y, arg1, ANIMAL), restrict(arg1, =, Crab)}> which, since the sense of the noun phrase only occurs in the restrict and was of form x on its own, puts Crab in place of x. Similar discharge rules exist to discharge other bind assumptions. Ultimately, we would get <BARK(Crab), {}> which is complete since no assumptions remain.

The twist to this is that pragmatic discharge rules may be invoked at any point during this computation of the conditional interpretation. There may be several ways of discharging a given assumption and several times at which they may be discharged. Some of these may fail. Pollack & Pereira discuss tactics for deciding when to discharge, although the details are not yet fully developed. What is interesting is that the *order* in which assumptions are discharged may determine which of a number of utterance readings will be obtained. One example of this is with quantifier scoping. Consider "Every boy likes a girl" which has readings in which the existential takes either wide-scope (the boys all like the same girl) or narrow-scope (the boys like possibly different girls). The conditional interpretation is <LIKES(y, x),

[6] Crab is Launce's dog in "*Two Gentlemen of Verona*".

{bind(y, $\forall$, BOY), bind(x, indef, GIRL)}>. It should be clear that discharge of the indefinite before discharge of other assumptions will create a new entity G, say, of type GIRL and will update the context with this. Since this is a constant it inevitably is not dependent on (i.e. not within the scope of) the universal. Alternatively, the assumption $< y$, {bind, $\forall$, BOY)}> could be discharged before the assumption associated with the indefinite. Pollack & Pereira allow discharge of a universal to select some subset of the outstanding indefinite assumptions and to discharge these too, giving them an existential interpretation within the universal's scope. This would give the other reading.

There is undoubtedly something very attractive about CANDIDE: it gives the option of early discharge (context consultation) but also allows decisions to be delayed if enough information has not yet been provided for a helpful discharge to be made. However, it is too early to fully judge this system. Pollack & Pereira need to investigate more fully what control strategy is needed, and when and on what basis assumptions should be discharged. What is interesting is that integrating semantic and pragmatic information in this way can offer solutions to the problems both of resolving certain structural ambiguities (e.g. certain PP attachment decisions) and of resolving scope ambiguities.

2.4.2 Hirst's Absity and Semantic Enquiry Desk

Hirst's system [Hirst 1987] deals with both structural and lexical ambiguity. [Hirst 1987] is reviewed in [Spärck Jones 1988] and the lexical disambiguation is critiqued in [Ogden 1989].

The system comprises a parser, *Paragram*, a semantic interpreter, *Absity*, a mechanism for lexical disambiguation, *Polaroid Words*, and a mechanism for structural disambiguation, the *Semantic Enquiry Desk*. The parser is deterministic with limited lookahead. Early use of semantic information is important as it can help to avoid making wrong decisions. Thus Paragram and Absity work in tandem: for each well-formed syntactic object detected by Paragram, a well-formed semantic object will be immediately created by Absity. This is achieved by having syntactic and semantic construction rules in one-to-one correspondence. As soon as a syntax rule combines constituents, a semantic rule combines the constituents' corresponding semantic objects. As the parser works from left-to-right through a sentence, words will be pulled in one at a time and intermediate syntactic and therefore also semantic structures will be created[7]. The semantic objects are elements of a frame-based language, Frail. Both the lexical disambiguation by Polaroid Words and the structural disambiguation using the Semantic Enquiry Desk make access to a knowledge base for reference resolution and general inference. Thus, Hirst's system, unlike CANDIDE, integrates all three types of processing: syntactic, semantic and pragmatic.

[7] Note that here I have been talking about the "base" syntactic rules. There are also "transformational" rules for handling "gapping" phenomena. These may be applied at any time and have no associated semantic rules. Further discussion of these is beyond the scope of this description.

Hirst tries to use the principle that as much disambiguation as can be done is done as soon as it can be done. This does occasionally lead to what Hirst terms "semantic garden paths": cases where a decision taken on current semantic information turns out to be wrong in the light of later information. This has some intuitive foundation to it since humans may be similarly misled (for example, many people initially select the wrong sense of "star" in "The astronomer married the star"). However, unlike humans, since Hirst uses a deterministic parser, his system cannot recover from these.

To handle lexical disambiguation, Absity wraps all the senses of a lexically ambiguous word into a "packet" called a Polaroid Word. Exact details need not intrude here ([Ogden 1989] is a good critique). The basic mechanism is that a Polaroid Word may slowly "develop", meaning that some of its candidate senses may be ruled out. A Polaroid Word has other Polaroid Words as its "friends" and may inspect its friends to help them and itself to develop. The information used for this comes mostly from selection restrictions and marker passing on the knowledge base.

A problem is that "friendship" has been defined only arbitrarily: it has no linguistic foundation to it. Furthermore, for reasons outside the scope of the present discussion, the system cannot handle passives (see [Ogden 1989]). It is also not clear what the relative usefulness of selection restrictions to marker passing is. All these problems make the system a little arbitrary.

I will now turn to structural ambiguity. Since an ordinary deterministic parser cannot back-up, it must detect structural ambiguity whenever it arises and decide immediately and irrevocably which alternative is better. Paragram and Absity will never attach anything to anything (*irrespective of whether ambiguity is possible or not*) without first asking the Semantic Enquiry Desk (SED) for permission. So, for example, once Paragram has parsed a PP and Absity has got semantic objects for the parse so far, the SED will be asked for help in deciding where to attach the PP. Chapter 4 mentions some of the sources of knowledge which Hirst's SED uses to make this decision (they include syntactic preferences, lexical preferences and the success or otherwise of references to the knowledge base). Chapter 4 also discusses the difference between Hirst's system and Haddock's in making these decisions. Basically Hirst has to wait until the PP has been recognised before using, for example, contextual knowledge to make the decision, whereas Haddock's system (and hence T42) can make the decision earlier than this (in some cases).

Hirst's system is impressive in its coverage of the cases of ambiguity which it can resolve. However, some of the methods it uses are somewhat arbitrary and the extent to which they contribute to disambiguation is not clear. It is also not clear how much Hirst's system would need changing for it to be able to handle texts rather than simple sentences: one result from [Ogden 1989] is that for texts marker passing becomes more important than selection restrictions and preferences.

::::

In this chapter I have characterised incremental systems as those that interleave different types of processing: syntactic, semantic and pragmatic. But to qualify as incremental, I insist that a system should allow feedback between these types of processing. This feedback might save the computation of unnecessary structures, but its real use is the leverage it gives in allowing the formulation of natural accounts of certain linguistic phenomena. I showed that this might be proven with a new argument from linguistics in favour of pragmatic intrusion, and also described two incremental systems which use their incremental designs for resolving lexical, scope and structural ambiguities. The rest of this thesis argues that incremental computation of presuppositions is also desirable. It does this by arguing first from the point of view of definite noun phrases and then extending this argument to other sources of presuppositions.

Chapter 3

T42: An Incremental System

It is one contention of this thesis that present NLP systems largely ignore contextual processing, and in so doing they cannot benefit from the advantages that may accrue from integrating syntactic, semantic and pragmatic processing, particularly the advantages relating to giving natural explanations to certain linguistic phenomena. T42, along with its precursors in the work of Mellish [1985] and Haddock [1987a, 1987b], and the other systems I described in Chapter 2 (Pollack & Pereira's CANDIDE and Hirst's Absity) are redressing this balance. These systems I have dubbed "incremental": they integrate syntactic, semantic and pragmatic processing to a much greater degree than other systems, while retaining a modular design. It should be noted, however, that complicated pragmatic processing such as the drawing of implicatures (e.g. [Hirschberg 1985]), speech act interpretation (e.g. [Cohen & Perrault 1979], [Allen & Perrault 1980]) and goal recognition (e.g. [Carberry 1983], [Pollack 1986]) still awaits integration with the rest of processing.

This chapter describes the modules that make up T42 and how they interact in processing a simple example. At this stage the example comprises only a definite noun phrase. More complicated definite NP processing is described in Chapter 4, where a more extensive example involving prepositional phrase attachment to definite NPs is given. The theoretical issues that underlie definite NP processing are discussed in Chapter 5. Then Chapter 6 shows how I have extended the system to handle indefinite NPs, verb phrases, prepositional phrases attached to things other than definite NPs, and less straightforward definite NP examples. However, the structure of T42, as described in this chapter, is not affected by these extensions. Information and control flow remain as described below; only the internal operations of one of the modules change slightly.

3.1 Outline System Description

T42 is based on the system described in [Haddock 1987a, 1987b]. This chapter concentrates on the foundations of the system as Haddock describes them, so the differences between his system and mine as far as the bulk of this chapter is concerned centre only on points of implementation detail. Differences emerge in Section 3.5 and in later chapters. First, Haddock describes only definite NP processing with his system. He does not show how the system might be extended to processing NPs as they occur in utterances of whole sentences. As Chapter 6 shows, I have further built on his foundations by extending the system so that it can process other parts of speech. Secondly, I make a distinction, which Haddock does not make (see Section 3.5), between the knowledge base and the discourse model, this being important for presupposition processing.

T42's design views semantic analysis as *constraint satisfaction*. [Mellish 1985] is the seminal work in this field. Mellish used constraint satisfaction to determine the referents of definite noun phrases. He incorporated this into a definite clause grammar parser and used it to process Physics exercises[1]. Haddock revived the idea of using constraint satisfaction for definite noun phrase interpretation, and came up with an elegant incremental parsing and interpretation system. I have taken this work and extended it to do sentential and discourse processing. Constraint satisfaction has also been used recently in systems for eliminating lexical ambiguity (see, e.g., [Mallery 1985] for determining the nature of the arguments to uses of copula verbs, [Stallard 1987], which is a proposal rather than a report of work done, and [Ogden 1989], which is a working system based on T42).

In Chapter 1 (Section 1.4) I gave a brief overview of T42's structure and its flow of control in processing an utterance. Here I remind you of that description by giving a slightly more detailed overview of processing a simple example. For the purposes of this overview, and for most of this chapter, I shall use the example of processing the definite NP "the sleepy porter". As I am giving an overview here I shall not go into the details of why this, as a definite NP, is processed in quite the way it is (see Chapter 5). My purpose here is to show the flow of information and the flow of control. Some of the details may not be clear at this stage. These will be explained in subsequent sections of the chapter and consolidated by a longer example in Chapter 4. I have repeated figure 1.1 here as figure 3.1, but I have numbered the arcs in the diagram to allow cross-referencing in the text.

Processing begins with the shift-reduce parser reading in the first word, "the" (1). The parser retrieves the lexical entry for "the", comprising both its category and its sense, from the categorial lexicon (2). The sense is in the form of a constraint. The parser keeps hold of the category ($\mathrm{NP}x_0/\mathrm{N}x_0$) and passes the sense ($\star$UNIQUE($x_0$)) to the constraint satisfier (3). The constraint satisfier pends this constraint. (I am being vague here: the constraint that is the sense of "the" is special, as indicated by $\star$, and no real action is taken on the basis of it yet). The constraint satisfier signals to the parser that it is 'happy' with the constraints so far (3).

The parser now reads in the second word, "sleepy" (1), and retrieves its lexical entry (2), i.e. category $\mathrm{N}x_1/\mathrm{N}x_1$ and constraint SLEEPY(x_1). The parser passes SLEEPY(x_1) to the constraint satisfier (3). (This is not a special

[1] In Mellish's system, constraint satisfaction is invoked only when processing definite noun phrases. In my extended system (Chapter 6) I use constraint satisfaction for all semantic analysis.

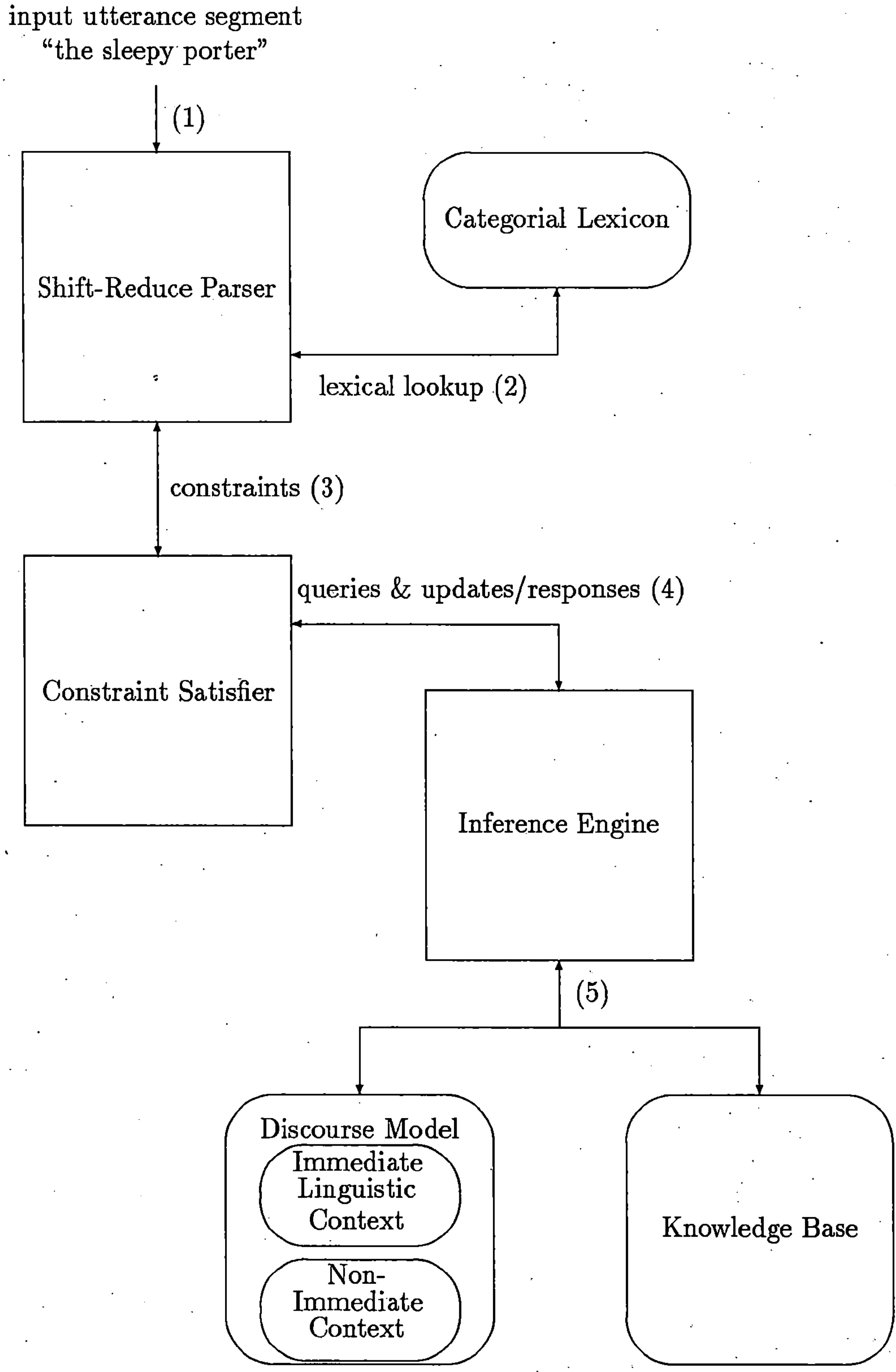


Figure 3.1: System Structure

constraint —no $\star$— and so can be actioned). The constraint satisfier asks the inference engine (4) to find instances of sleepy things in the discourse model, possibly also making use of knowledge from the knowledge base. This the inference engine does (5). Assume that the discourse model only knows of one sleepy thing, labelled P. P will be passed back to the constraint satisfier (4). This satisfies the constraint satisfier, which signals to the parser that the parse may continue (3).

Before going onto the next word, the parser now tries to combine the two categories it has held onto, the one for "the" and the one for "sleepy". This is successful, giving NPx_0/Nx_1. In combining these, i.e. in recognising that "the" and "sleepy" can form the beginning of a constituent, the parser generates a new semantic constraint. The constraint it generates is EQUAL(x_1, x_0) and this is passed to the constraint satisfier (3). The constraint satisfier interprets this as saying that x_0 and x_1 refer to the same thing. And, since it already knows that x_1 refers to P, it now knows that x_0 refers to P too. No conflict is caused by this and so the constraint satisfier signals to the parser that parsing may continue (3).

The parser reads in "porter" (1) and gets its lexical category (Nx_2) and constraint (PORTER(x_2)) from the lexicon (2). The constraint is dispatched to the constraint satisfier (3). The constraint satisfier requests instances of porters from the discourse model and these are returned ((4) and (5)). Assume that this also returns P, i.e. the discourse model knows about only one porter, P. Retrieval of this entity satisfies the constraint satisfier and it signals this to the parser (3).

The parser now attempts to combine the latest category (Nx_2) with that of the partial phrase it already has (NPx_0/Nx_1). This is possible to give a category NPx_1 to the whole phrase "the sleepy porter". In doing this, a new constraint is generated, EQUAL(x_2, x_1), which is passed to the constraint satisfier (3). Also, whenever the parser recognises an NP (or an S) it informs the constraint satisfier of this. The constraint EQUAL(x_2, x_1) tells the constraint satisfier to check whether the values it has for x_1 and x_2 are equal. They are, since both are P. Then, since it has also been told that an NP has been recognised, this tells the constraint satisfier to action any constraints with $\star$s on them. In this case, this means checking whether x_0 has a unique value: x_0 is equal to x_1, which has the value P, hence x_0 does have a unique value. The constraint satisfier thus signals that it is 'happy' to the parser (3).

There are no more words to read in, the parser has detected a NP and the constraint satisfier has indicated that no constraints are violated: processing has completed successfully. The constraints the constraint satisfier has accumulated and also the discourse model entities it retrieved to satisfy those constraints will give some kind of logical form for the input phrase. In the case of processing an utterance and not just an isolated definite NP, some parts of the logical form will be "new" and other parts will be already "given" (see Chapter 6). The "new" bits will be passed from the immediate linguistic context to the non-immediate context before processing of the next utterance begins.

I have suppressed lots of details in this overview. In particular, I have not said anything about what happens when constraints are violated or when the parser cannot combine two consecutive phrase markers. These details are brought out in the context of the example in Chapter 4. In the remaining sections of this chapter I shall describe the internal workings of the system modules, taking the categorial lexicon, the shift-reduce parser, the constraint satisfier and the inference engine, discourse model and knowledge base in turn.

3.2 Categorial Lexicon

The grammar formalism used in this system is *categorial grammar* (see, e.g. [Steedman 1985, 1987]). Categorial grammars, it is claimed, are especially well suited to incremental processing.

There are no phrasal grammar rules in such a grammar. Rather, the grammar is specified by a *categorial lexicon* and a number of general *combination rules.* The categorial lexicon is just a list of lexical entries, in which an entry's category may be either primitive or derived from primitives. A word or phrase's category indicates what other categories the word or phrase can occur and combine with. The primitive categories in the grammar used here are sentence (S), nominal (N) and noun phrase (NP). Derived categories are built up from the primitive categories, the operators / and \ and parentheses. If a word or phrase is of derived category X/Y (where X and Y are categories), then it will combine with something of category Y to its *right* to form a phrase of category X. For example, "the" has category NP/N, i.e. "the" combines with an N to its right to form an NP. If a word or phrase is of derived category $X\backslash Y$ (where X and Y are categories), then it will combine with something of category Y to its *left* to form a phrase of category X. For example, "sleeps" has category S\NP, i.e. "sleeps" combines with an NP to its left to form an S.[2] A category of the form Y/Z or $Y\backslash Z$ (where Y and Z are categories) is to be thought of as a function over Zs. For example, a transitive verb, such as "likes", has the category (S\NP)/NP, which shows that "likes" is a function from noun phrases, NP, to intransitive verbs, S\NP, which themselves are functions from NPs into sentences, S.

There are then the following combination rules (where $X, Y,$ and Z are categories)[3]:

- Forward application (⇒apply): $X/Y + Y \rightarrow X$
  If immediately to the right of something of category X/Y there is something of category Y, then rewrite these to category X
  e.g. "the" is category NP/N, "porter" is category N, so these rewrite to category NP ("the porter").

[2] It is possible to add the usual syntactic features, e.g. for number agreement, to the grammar. I have used a few rudimentary ones in the system but will not bother to show them in this thesis.

[3] There is a corresponding set of backward application and composition rules (e.g. backward application (⇐apply): $Y + X\backslash Y \rightarrow X$).

- Forward composition (⇒compose): $X/Y + Y/Z \rightarrow X/Z$
  If immediately to the right of something of category X/Y there is something of category Y/Z, then rewrite these to category X/Z
  e.g. in subject position, "Orsino" is category S/(S\NP), "likes" is category (S\NP)/NP, so these rewrite to category S/NP ("Orsino likes").

- Forward composition2 (⇒compose2): $X/Y + (Y/W)/Z \rightarrow (X/W)/Z$
  If immediately to the right of something of category X/Y there is something of category $(Y/W)/Z$, then rewrite these to category $(X/W)/Z$
  e.g. in subject position, "Orsino" is category S/(S\NP), "gives" is category ((S\NP)/NP)/NP, so these rewrite to category (S/NP)/NP ("Orsino gives").

As [Steedman 1985] and [Haddock 1987b] point out, the two composition rules are particularly useful in an incremental system since they allow the grammar to give phrase markers to phrases that other grammars would not recognise as constituents. This can be seen in the examples "Orsino likes" and "Orsino gives" above: categorial grammar does not have to determine whether the verbs have suitable object NPs before being able to assign phrase markers to these. The phrase markers it assigns reflect the fact that object NPs are needed to complete the sentence. This is what makes categorial grammar the natural choice for T42.[4]

In addition to the combination rules, there are rules for *type-raising* categories. Type-raising can change a category from being an argument into being a function (e.g. it might take a primitive category and turn it into a derived category, i.e. make it a function over other categories). Thus, type-raising may be used to enable syntactic categories to combine earlier than they otherwise would have. This does not affect the success of the parse, merely the type and order of combinations that find the parse.

Subject type-raising, for example, raises category NP to S/(S\NP), i.e. to a function over verb phrases (S\NP). The rule should only be applied to NPs in subject position. The type-raised category combines with verb phrases to its right to return sentences. This would seem to be intuitively reasonable. What is more, it can immediately apply to any verb to its right to return constituents for incomplete phrases such as "Orsino likes" and "Orsino gives". Ordinarily, without such a rule, a parser would have to wait until the rest of the input has been recognised as a verb phrase before backward applying the verb phrase to the subject NP. *Topic type-raising* raises a category X to a category S/(S/X). For example, in "Apples, Orsino likes!", if "Apples" is topic type-raised from category NP to S/(S/NP), this forward applies with "Orsino likes", category (S/NP), to give S. Note that for English one needs restrictions on which categories can be topic type-raised (see [Steedman 1987]). Other type-raising rules allow prepositional phrase attachment to nouns and sentences. Nouns (N) are raised to type N/(N\ N). Sentences (S) are raised to (S/((S\NP)\(S\NP))). Prepositions have categories (N\ N)/NP and ((S\NP)\ (S\NP))/NP; the first of these

[4] This is not to say that the same things could not be achieved using traditional phrase-structure grammars: they could, but less easily and less naturally.

can compose with a type-raised nominal and the second can compose with a type-raised sentence to give N/NP and S/NP.

Since type-raising facilitates immediate category combination, it is very useful in an incremental system, where we want to do as much as we can before going on to later parts of an utterance. We shall see that the parser is designed to prefer left-branching parses, where categories are combined as soon as possible, to right-branching ones (which use backward-looking rules).

Categorial grammar when augmented with rules of composition and type-raising is referred to as *combinatory categorial grammar.* These extra rules allow us to handle "discontinuous constituents" or "extraction" phenomena (e.g. "Apples, Orsino likes!") without resorting to "transformations" or "slash feature passing". Unfortunately, these rules multiply the number of syntactic parses for a sentence. For example, "Orsino saw Viola" can have the following two derivations:

```
(a) Orsino saw        Viola    (b) Orsino saw        Viola
    ------ ---------  -----        ------ ---------  -----
    NP     (S\NP)/NP  NP           NP     (S\NP)/NP  NP
    ----->raise                           --------------->apply
    (S/(S\NP)                                    S\NP
    --------------->compose        ---------------------<apply
          S/NP                               S
    ---------------------->compose
                     S
```

Parse (a) is a left-branching structure and uses type-raising and composition to makes use of the subject NP immediately; parse (b) is a right-branching structure which has one less syntactic operation but does not use the subject NP until its final backward apply. In general, the rules will be able to assign not only both the left-branching and right-branching structures, but all mixtures in between. The problem with this is that each of these analyses will be semantically equivalent. For this reason, this is called "spurious ambiguity". Avoiding spurious ambiguity is left to the parser, which is the subject of the next section.

3.3 Shift-Reduce Parser

To rule out spurious ambiguity, a sophisticated parser is needed. However, since parsing is not the focus of my work, I chose to use a shift-reduce parser. While my parser might not rule out spurious ambiguity properly, it is extremely simple and is otherwise well suited to categorial grammar parsing. Basically at any one step in a shift-reduce parser, another item may be shifted from the input buffer onto the top of a stack, or consecutive items on the top of the stack may be reduced to another category (i.e. the top n items can be recognised to constitute a higher level category). In 'conventional' grammars with phrase structure rules, a reduction occurs on recognising that the top n items of the stack match the right-hand side of a rule. With categorial grammar, reductions are simpler: they only ever involve the top two items of the stack. A reduction occurs when these two items can be combined according to one of the combination rules. For example, if the stack top is of category Y and the item below it is of category X/Y, then forward application allows their combination and replacement by category X.

The next subsection describes the control strategy of the parser, with particular regard to ruling out spurious ambiguity; this is followed by a subsection which describes how the parser generates the constraints which it sends to the constraint satisfier.

3.3.1 Parser Control Strategy

The *control strategy* of a shift-reduce parser must determine what to do in the face of *conflicts*, i.e. when faced with a choice of next possible operations. The parser might have to choose whether to do a reduction, one of a number of applicable type-raisings or, in the case of a categorially or lexically ambiguous word, one of a number of shifts. Dispreferred options may either be discarded or put onto an agenda of outstanding options.

The control strategy of the parser for T42 must also try to rule out spurious parses. This it does by trying to find only the left-branching parse. In essence this means it will have a *reduce-first strategy*, where reductions are preferred to shifts. Type-raising will also be preferred to shifting as this can increase the chance of doing a reduction as the next operation. A reduce-first strategy has the effect of preferring to close 'open' constituents as soon as possible. This fits nicely into incremental processing as it quickly forms constituents about which the other modules may voice an opinion.

I use the following conflict resolution rules to achieve this reduce-first strategy:

1. If the parser can do a reduction, it does. Any possible type-raisings are put onto the agenda. Shifts are disallowed.

2. If no reductions are possible but the parser can do a number of type-raisings, one of these is chosen arbitrarily. Other type-raisings go onto the agenda and again shifts will be disallowed.

3. If neither reductions nor type-raisings are possible, the parser shifts. In the case of (categorially or lexically) ambiguous words, one shift will be arbitrarily chosen and the other(s) will be put onto the agenda.

4. If no reductions, raisings or shifts are possible, or if the constraint satisfier rejects the current analysis, then this parse is discarded and work restarts on one of the partial parses on the agenda.[5]

[5] A minor 'tweak' to these rules is that even if there are type-raising rules which can be applied to the category on top of the stack, a one word lookahead to the category or categories of the next word checks whether this type-raising will be of any use; a type-raising rule is applied only if this is so.

Using these particular rules is little more than a hack. I do not know if all spurious parses will no longer be found and, perhaps more seriously, the rules are over-restrictive: certain non-spurious parses will be lost[6]. Overcoming these problems is beyond the scope of this dissertation. Haddock used a form of chart parser, as described in [Pareschi & Steedman 1987], which was supposed to overcome these problems[7].

As the example at the start of the chapter indicated, the parser is responsible for passing constraints to the constraint satisfier. How these constraints are produced is the subject of the next section.

3.3.2 Semantic Constraints

In adding semantics to categorial grammar the *rule-to-rule hypothesis* could be used, just as it often would in a more conventional system: each syntactic rule is paired with a corresponding semantic rule which says how to form the semantics of the phrase from the semantics of the constituents of the phrase.

In categorial grammar, "syntactic categories directly reflect the semantics of the entity in question" [Steedman 1987, p.32][8]. So, the category of "likes", (S\NP)/NP, tells us not only that the type of its syntactic function is a function from NPs to intransitive verb phrases which are themselves functions from NPs to sentences; it also tells us that the semantics of "likes" is a function whose type is a function from entities (the object) to functions from entities (the subject) to truth-values. The semantics of the combination rules will reflect this translation into semantic functions. For example, forward application $(X/Y + Y \rightarrow X)$ has the semantics $F(y)$ where F is the semantic function corresponding to X/Y and y is the semantic object corresponding to Y. Forward composition $(X/Y + Y/Z \rightarrow X/Z)$ has the semantics of $\lambda x[F(G(x))]$, the composition of F and G where F and G are the semantic functions corresponding to X/Y and Y/Z respectively.

Following Haddock [1987a, 1987b], the method of doing semantics in T42 has some similarities with the above, but is also a little different since T42 uses constraints, not semantic functions. These constraints come from the lexical entries for words and can also be generated by syntactic operations of the parser.

The lexical semantics of a word sense will be given as a list of constraints. As the example at the start of the chapter indicated, the constraint satisfier takes these constraints and, with reference to the discourse model, attempts to instantiate the variables with constant values. For example, we might have the following entries in the lexicon[9]:

[6] An example is that the parser gets only one analysis for sentences of the form "I believe I like coffee and I like tea." It gets the analysis "[I believe I like coffee] and [I like tea]" but not "I believe [[I like coffee] and [I like tea]]".

[7] Described as a "lazy chart parser", this parser, it was claimed by its authors, avoided spurious ambiguity without losing any real parses. However, [Hepple & Morrill 1989] have claimed that Pareschi & Steedman's parser is not complete.

[8] This is a more restricted version of the rule-to-rule hypothesis and sometimes referred to as "type-driven translation".

[9] I ignore all complexities of the actual lexical semantics one might need.

word	*category*	*meaning*
"porter"	Nx	PORTER(x)
"sleepy"	N/Nx	SLEEPY(x)
"the"	NPx/Nx	$\star$UNIQUE(x)[10]

In this we can see that constraints are expressed in terms of variables that are used to label primitive categories within the category of the word. So, for example, the category Nx indicates that x will be the entity described by the N. The semantics can then be expressed as PORTER(x), where x is the argument to the predicate.

We also need to capture the semantics generated by the combination rules. What we do here must reflect the functional composition mentioned earlier. Following [Haddock 1987a, 1987b], I do this by generating a special constraint between the variables that label categories which get eliminated by a rule application: when we use forward application to reduce X/Yy_0 and Yy_1 to X, where we see y_0 and y_1 labelling the two instances of syntactic category Y, then we generate a constraint that y_0 and y_1 must denote the same object, i.e. EQUAL(y_1, y_0). More generally, if two categories 'wipe each other out' in the application of a combination rule, then we generate a constraint saying that the denotations of their variables must be equal.

The way that syntactic and semantic operations are combined can be illustrated by looking at "the sleepy porter" in more detail than was done before. When a word is shifted, its semantics are passed to the constraint satisfier. The effect of this will be considered later. Note that variables get renamed on shifting.

We begin by shifting "the", and sending off its semantics:

$$\text{Stack: } \boxed{\mathrm{NP}x_0/\mathrm{N}x_0} \qquad\qquad \Longrightarrow \star\texttt{UNIQUE}(x_0)$$

Then we shift "sleepy" and dispatch its semantics. The stack will be as follows:

$$\text{Stack: } \boxed{\begin{array}{l}\mathrm{N}x_1/\mathrm{N}x_1\\ \mathrm{NP}x_0/\mathrm{N}x_0\end{array}} \qquad\qquad \Longrightarrow \texttt{SLEEPY}(x_1)$$

(where the top of the stack is higher on the page). A reduction using forward application is possible, to give:

$$\text{Stack: } \boxed{\mathrm{NP}x_0/\mathrm{N}x_1} \qquad\qquad \Longrightarrow \texttt{EQUAL}(x_1, x_0)$$

Here the system recognised that the Nx_1 of the stack top could be 'equated to' the Nx_0 in the item below the stack top: this is what permits the reduction by forward application. When this happens, the *variables* of these

[10] This is actually a pragmatic constraint and not a semantic one, as signalled by the $\star$. Its special behaviour has been mentioned earlier and will be properly exemplified in Chapter 4.

equal, 'factored out' categories are also set equal, i.e. the system generates a constraint saying that these variables are equal, in this case $\texttt{EQUAL}(x_1, x_0)$. This is sent to the constraint satisfier. Here, it forces whichever sleepy things we find in the extension of x_1 to be the same as the things we find satisfying the constraints of "the" (variable x_0).

"porter" is now shifted onto the stack and its semantics are sent off:

$$\text{Stack: } \begin{array}{|l|} \text{N}x_2 \\ \text{NP}x_0/\text{N}x_1 \\ \hline \end{array} \qquad\qquad \Longrightarrow \texttt{PORTER}(x_2)$$

Forward application leaves NP on the stack and will create the constraint $\texttt{EQUAL}(x_2, x_1)$:

$$\text{Stack: } \begin{array}{|l|} \text{NP}x_0 \\ \hline \end{array} \qquad\qquad \Longrightarrow \texttt{EQUAL}(x_2, x_1)$$

This completes the parse successfully (there is no more input to read in and we are left with just an NP on the stack). If we look at the semantic constraints that were generated during the parse:

i.e. $\star\texttt{UNIQUE}(x_0)$	$\texttt{SLEEPY}(x_1)$	$\texttt{EQUAL}(x_1, x_0)$
$\texttt{PORTER}(x_2)$	$\texttt{EQUAL}(x_2, x_1)$	

we see that x_0 must now not only be unique, it must be a sleepy thing (since it is equal to x_1) and must be a porter (since it is equal to x_1, which is equal to x_0).

The constraints generated during parsing are sent one at a time to the constraint satisfier. I will now describe what the constraint satisfier does with them.

3.4 Constraint Satisfier

Constraint satisfaction is a well-established technique within Artificial Intelligence. Given a set of variables, each associated with a set of candidate values (a domain), and a set of constraints which restrict the assignments that may be made to specified subsets of the variables, the *Constraint Satisfaction Problem* (CSP) is the problem of finding the sets of assignments of values to variables that satisfy all the constraints. The set of values that may be consistently assigned to a particular variable (i.e. consistent with respect to the constraints) is called the variable's *satisfaction set.* The satisfaction set will be some subset of the variable's candidate values (i.e. its domain).

A variety of algorithms have been proposed to handle CSPs (see [Shanahan & Southwick 1989]). I have implemented a simple breadth-first approach, whose operation I will illustrate with the example "the sleepy porter". From the previous section, we know that the following constraints are sent to the

constraint satisfier by the parser in the following order:
1. $\star$UNIQUE(x_0)
2. SLEEPY(x_1)
3. EQUAL(x_1, x_0)
4. PORTER(x_2)
5. EQUAL(x_2, x_1)
I will ignore constraints (1) and (3) which concern the variable x_0 used in the special constraint labelled $\star$ (1). Testing for uniqueness is not material to the discussion of the way the constraint satisfier works.

The second constraint SLEEP(x_1) introduces a new variable into the CSP with a constraint on that variable. The satisfaction set of x_1 will be all objects in the discourse model which are recorded as being sleepy.[11] In the introduction to this chapter I assumed there was only one such object. This does not illustrate constraint satisfaction well so here I will assume that there are three such objects P, Q and R. Thus the satisfaction set of x_1 is {P, Q, R} and the simple CSP consisting of one variable x_1 and one constraint SLEEPY(x_1) has three solutions <P>, <Q> and <R>.

Ignoring constraint (3), constraint (4) is the next significant constraint here. It introduces a new variable x_2 and a new constraint PORTER(x_2). The constraint satisfier thus retrieves from the discourse model all x_2 that are porters. I will assume that there are two such objects, O and P, and so x_2 gets a satisfaction set of {O, P}. This gives $3 \times 2 = 6$ solutions, <P, O>, <P, P>, <Q, O>, <Q, P>, <R, O> and <R, P>.

Now the fifth constraint EQUAL(x_2, x_1) arrives and is actioned. It introduces no new variables but does prune the solution space to <P, P>, the only case where x_2's value equals x_1's value, and cuts the satisfaction set of x_2 to {P} and of x_1 to {P}. It is significant that I immediately use this constraint to prune the solution space. Suppose another variable x_3 is introduced whose satisfaction set is {A, B, C, D}. If the constraint satisfier had not done this pruning the current solution space would have contained $6 \times 4 = 24$ possible solutions, i.e. <P, O, A>, <P, O, B>, <P, O, C>, <P, O, D> <P, P, A>, ... But because the solution space *was* pruned to only one solution the new solution space after introduction of x_3 will contain only $1 \times 4 = 4$ possible solutions, i.e. <P, P, A>, <P, P, B>, <P, P, C> and <P, P, D>.

It may seem fortuitous in this example that pruning reduced the solution space to a single item <P, P> after constraint (5), but my claim is that this would be quite often the case in natural language processing. We can think of a satisfaction set as containing competing referents for a word or phrase and since, intuitively, human language processors do not entertain large numbers of readings concurrently for very long (i.e. language ordinarily furnishes us with enough constraints and preferences to keep the number of readings small) perhaps this degree of pruning is not so unusual. Solving CSPs is, in general, NP-complete, i.e. we believe the problem will take time exponential in the number of variables to solve. While this characterises

[11] I have simplified here: we really ought not to search the *whole* discourse model, only the most "salient" parts of it.

worst-case behaviour of CSP algorithms, I believe that the application of this technique to language processing will almost certainly not exhibit worst-case behaviour. This intuition means that my simple breadth-first algorithm has proved adequate (although it must be admitted that no very complicated examples have been processed using T42).[12,13]

If at any time one or more satisfaction sets becomes empty, then the CSP is *unsatisfiable*, i.e. there is no solution that satisfies all the constraints. In this case, the constraint satisfier would signal to the parser that this was so and request that an alternative analysis be tried.

You will see from the above description that the constraint satisfier needs to access a set of constraints, a set of variables and the variables' satisfaction sets while processing an utterance. I refer to this information as the *immediate linguistic context* (ILC). The ILC is part of the *discourse model* (see next section). It contains the information accumulated while processing earlier parts of the the current utterance. The rest of the discourse model, referred to as the *non-immediate context* (NIC), contains other contextual information. Once the utterance has been successfully processed, information representing the chosen reading for the sentence is moved from the ILC into the NIC. The ILC is thus empty when processing of the next utterance begins.

It should be made clear that these labels ("discourse model", "immediate linguistic context" and "non-immediate context") are nothing more than labels for parts of the system. Their suggestiveness of models of memory is perhaps dangerous: it has not been an objective of this work to contribute to this. However, the distinction between the ILC and the NIC needs to be explained further. This I do in the next section.

3.5 Inference Engine, Discourse Model and Knowledge Base

This thesis has nothing to say about models of memory and corresponding structuring of knowledge bases. As T42 requires a knowledge base, one has been provided, but it is intended to do no more than offer the minimum necessary support for T42's primary processing and has been kept as simple as possible. Almost none of the many distinctions that could have been drawn have been drawn. However, one rather gross distinction is made, and this proves to be of critical importance. The knowledge T42 uses can be thought of as split into two: general *world knowledge*, held in the *knowledge base*, and *contextual knowledge*, the latter represented as a *discourse model.*

T42 treats general world knowledge held in the knowledge base as if it is all mutually believed by itself and the user. The basis of this mutual belief would

[12] I have also optimised it a bit so that in extending the depth of the search by a level, a constraint is not repeatedly checked on the same values. These details do not seem important enough to include in full.

[13] Mellish and Haddock, for reasons that are not clear, both use a constraint propagation approach [Freuder 1978], [Shanahan & Southwick 1989].

be what Clark & Marshall call "community membership" [Clark & Marshall 1981]. Since these beliefs are assumed to be mutual, T42 can use any of them it wants in interpreting a user utterance. I recognise that more realistically distinctions would have to be drawn. For example, there are T42's private beliefs, there are things T42 believes are mutual and there are things T42 believes the speaker believes are mutual but to which T42 does not accede, and so on. World knowledge in T42 primarily consists of rules representing the usual type- or generalisation hierarchy[14]. These say, e.g., that porters are humans, humans are mammals, mammals are animals, animals are living-objects, and so on[15]. Other rules represent more general facts, e.g. cars have steering wheels. Little work has been put into encoding any of this knowledge. Only knowledge essential to running particular examples has been tackled and even this has been encoded in a simplistic form. The problems of uncertain or merely plausible knowledge, for example, have been ignored.[16]

Other than this general world knowledge, there is *contextual knowledge.* Contextual knowledge, I have assumed, is knowledge that more directly pertains to the *current discourse*, rather than background, general knowledge arising from community membership. I assume that this contextual knowledge is more salient than other knowledge in some way, although different degrees of salience *within* contextual knowledge have been ignored. I am assuming that all contextual knowledge is also mutual. Note that I say it is "mutual"; I do not say that it is mutually believed or mutually known. I am going to suggest that satisfaction of presuppositions does not have anything to do with propositional attitudes held about contextual knowledge. This is just knowledge that is being 'entertained' by virtue of having arisen in the discourse.

Contextual facts are divisible into the linguistic and the non-linguistic. *Non-linguistic contextual knowledge* is mutual through "physical copresence" [Clark & Marshall 1981]. This contextual knowledge gives properties about objects evoked by the physical discourse setting, e.g. that there is a black cat in the conversational participants' visual fields. *Linguistic contextual knowledge* is mutual through "linguistic copresence" [*Ibid*]. This contextual knowledge gives properties about objects that were introduced by previous utterances in the discourse, i.e. facts learnt from the discourse itself.

I will use the phrase *discourse model* to refer to T42's representation of contextual knowledge as a whole, i.e. it covers non-linguistic contextual knowledge and both immediate and non-immediate linguistic contextual knowledge. I shall treat the discourse model as if it is organised as a set of objects evoked by the discourse with properties also evoked by the discourse about those objects. This is close to Webber's use of "discourse model" to refer to "...a structured collection of entities, organised by the

[14] Presently this is restricted to being a tree and not a network.

[15] A program 'compiles' these rules into an alternative form to allow an efficient implementation of mutual exclusion tests, e.g. if something is a porter it cannot also be a mouse or an event.

[16] Should T42 be interfaced to an application program, general world knowledge would have to be supplemented with application and task specific *domain knowledge.*

roles they fill with respect to one another, the relations they participate in, and so on." [Webber 1981, p.283]. In defining "discourse model", Webber takes the *speaker's* point of view; the discourse model is what the speaker wants to communicate; some part of it will already have been communicated and the speaker can take this as mutual. But I am taking the *hearer's* point of view: the discourse model, for the hearer, is the set of entities and their properties conveyed to him in the discourse so far. I am not assuming that the hearer necessarily believes the entities exist in, e.g., the 'real' world, nor that he believes the properties to be true in the 'real' world. As I have said, the discourse model is 'mutual' in the sense of containing entities and their properties evoked by the discourse (previous utterances or physical setting). If the speaker uses a referring expression, the hearer will attempt to identify a discourse model entity. It does not matter much whether he believes the entity 'really' exists or whether he believes the properties 'really' do hold about the entity.[17] Of course, in a more sophisticated system these distinctions (what the hearer believes, what he believes is mutually believed, etc.) would be important in determining what T42 itself might say in a discourse. But in T42 as it stands what is important is that these are mutual in the sense of having arisen in the discourse and thus being things that can be referred to.[18]

As was mentioned in the previous section, from an implementation point of view the discourse model is split into two: the immediate linguistic context (ILC) and the non-immediate context (NIC). The ILC holds information gleaned from the utterance currently being processed. Its updates come from the constraint satisfier as processing of the utterance proceeds. It is thus a 'scratchpad' for the constraint satisfier. As such, the information it holds is in the 'format' of a constraint satisfaction problem (i.e. constraints on variables with satisfaction sets).[19]

The non-immediate context (NIC) holds information from previous parts of the discourse and, in principle, also holds items of non-linguistic context.[20] This information is held in the 'format' of formulas of clausal form logic. The principle way in which the NIC is updated is that, at the end of processing the current utterance, information is transferred from the ILC to the NIC.[21]

[17] It is this that allows one to felicitously talk of perpetual motion machines, Father Christmas and other fictional, mythological and even impossible objects.

[18] I thank Karen Spärck Jones for making me think along these lines, i.e. that the conversational participants need not have any particular propositional attitudes towards the contents of the discourse model, they need only be 'entertaining' these things because they have arisen in the discourse. The idea has a precursor in Stalnaker's characterisation of a context. With reference to a proposition being in context, he says: "This does not imply that the person need have any particular mental attitude toward the proposition, or that he need assume anything about the mental attitudes of others in the context." [Stalnaker 1972, p.387]

[19] Further, when a new variable is introduced into the constraint satisfaction problem and the constraint satisfier retrieves an initial satisfaction set for the variable, it will retrieve entities from both the ILC and the NIC. If it were not able to also consider entities in the ILC, T42 would not be able to handle cases of intra-sentential reference.

[20] Since the NIC can, in principle, contain items of the immediate non-linguistic context, to refer to it simply as non-immediate context is a slight misnomer.

[21] In experimenting with T42, other entities and properties are entered into the NIC by

Since the ILC is held in a format more suited to the constraint satisfier, but the NIC is held as clausal form logic, the transfer from the ILC to the NIC involves some change of format. Examples of this change will be given throughout the thesis. However, it is important to point out that this rewriting from one format to another is no more than a change of data structure. There is no sense in which the ILC and NIC contain information with different model-theoretic interpretations. In other words, although the ILC contains information held in a constraint satisfaction format, this can be viewed as an unusual way of representing the same sort of clauses as are held in the NIC.[22]

Thus the discourse model as a whole (both the ILC and the NIC) contains a set of propositions. This set will determine a set of possible worlds: those that are consistent with the propositions. The idea of drawing a distinction between contextual knowledge (the discourse model) and possible worlds is also found in [Stalnaker 1972]. Since possible worlds and contexts together determine truth-values, they could in principle be merged (i.e. part of a context would be the possible world in which the sentence is uttered). However, Stalnaker sees a "functional difference" between contexts and possible worlds, which manifests itself in giving accounts of presuppositions and of pragmatic ambiguity. In particular, the propositions in the context (discourse model) need not be true in the 'real' world. Hence, as I shall explain later, presuppositions also need not be true in the 'real' world, nor need their falsity in the 'real' world affect our ability to obtain a proposition which can then be assessed for truth in a number of possible worlds. This should enable us to give an account of, e.g., deception, inaccurate reference, fiction, etc.

Access to the knowledge base and the discourse model is made via the inference engine.[23] This is simply a rule-based interpreter. Queries from the constraint satisfier are answered by deductive retrieval, i.e. instantiating variables and backward chaining through rules until ground clauses that satisfy requests and sub-requests are found; variable instantiations are returned as answers. Updates initiated by the constraint satisfier to the ILC may trigger additional updates caused by the invocation of forward-chaining rules (see Chapter 7). The implementation is based on [Charniak et al 1980].

::::

hand prior to utterance processing, to represent information that might have arisen as a result of things that T42 has already uttered or through being evoked by the physical discourse setting.

[22] This would appear to mean that there is no difference between the ILC and the NIC other than that each is stored in a way that is more amenable to the processing done on it. This is not entirely the case: Chapter 9 shows the distinction between the two is used to achieve a notion of "accessibility" of candidate referents.

[23] If the inference engine wishes to do reasoning with the contents of the ILC it has to first translate the constraint satisfaction format data into clausal form logic.

This chapter discussed the details of each module of T42 in the context of a very simple example, "the sleepy porter". In the next chapter a more complicated example is presented which will demonstrate how one linguistic problem, that of prepositional phrase attachment, can be given a natural treatment in a system such as T42. The focus at this stage remains on definite NPs (and attachment of PP to such NPs).

Chapter 4

An Example Using T42

This chapter continues to look at the basic version of T42 outlined in the previous chapter and so restricts itself to the processing of a definite noun phrase example. The main aim is to consolidate the description of processing in T42 given in the previous chapter. Since this means that I am still explaining the simple system based on Haddock's work, I have chosen to base my example on Haddock's. The example involves deciding whether to attach a prepositional phrase to a definite noun phrase. It thus illustrates how incremental processing can give leverage in resolving structural ambiguities. The example differs only slightly from Haddock's in that his involves processing "the rabbit in the hat" in a context containing three rabbits of which only one is in a hat, one is in a box and one is not in anything. I process "the portrait in the casket" in a context containing three portraits, one in a casket, one in a tub and one not in anything. But some people have remarked that it was not immediately clear how a casket with something other than a portrait in it (or a hat with something other than a rabbit in it) would affect the example. So I have included another casket which has a book in it in order to illustrate this facet.[1] I have also put the example into context by giving an overview of the issues surrounding prepositional phrase attachment, and conclude the chapter with a discussion of the uniqueness constraint on definite NPs.

4.1 Prepositional Phrase Attachment

Prepositional phrases can be a source of structural ambiguity. A few examples are given below:

(1) a. "Iago saw the man with the telescope."
b. 'Iago saw [the man with the telescope].'
c. 'Iago saw [the man] [with the telescope].'

[1] I have avoided rabbits because I cannot draw them. My example is inspired by Shakespeare's *Merchant of Venice* in which Portia's suitors must pass a test involving three caskets. In the play, only one casket contains a portrait, there's no book and there's no tub — my apologies to Shakespeare for this and other examples in this dissertation!

Gloss: Did the man have the telescope (b) or did Iago use the telescope to see the man (c)?

(2) a. "Put the block in the box on the table."
b. 'Put [the block in the box] [on the table].'
c. 'Put [the block] [in the box on the table].'
Gloss: Are we to move the block from the box to the table (b) or are we to move the block to the box which is on the table (c)?

(3) a. "The cupboard near the door with the brass handle is made of mahogany."
b. 'The cupboard [near the door with the brass handle] ... '
c. 'The cupboard [near the door] [with the brass handle] ... '
Gloss: Is it the door which has a brass handle (b) or the cupboard which has the brass handle (c)?

All three of the (a) sentences have at least two readings ((b) and (c)) which differ in the category of the parent of the prepositional phrase, i.e. what the prepositional phrase is "attached to". In the first two examples, the PP may be attached to the object NP (reading (b)) or the verb (reading (c)). The third example shows an example in which the PP may be attached to one of two NPs. In general, PPs may be attached to the verb, the object NP or any other verbal complement (e.g. an adjectival phrase) or an NP in a preceding PP.

That any simplistic 'solution' will not work is easily demonstrated with the following two examples:

(4) a. "Viola wanted the dress on the rack."
b. 'Viola wanted [the dress on the rack].'
c. 'Viola wanted [the dress] [on the rack].'

(5) a. "Viola positioned the dress on the rack."
b. 'Viola positioned [the dress on the rack].'
c. 'Viola positioned [the dress] [on the rack].'

Intuitively, the preferred interpretation of (4a) is (4b), i.e. the PP is attached to the object NP, while the preferred interpretation of (5a) is (5c), i.e. the PP is attached to the verb phrase. This means that any simple syntactic preference rule will be insufficient: for example, if a syntactic preference tells us that in the absence of semantic or pragmatic information we would prefer VP-attachment to NP-attachment in the examples above[2], we also need to be told when to use the preference and when to ignore it.

Another class of proposed 'solution' makes use of what Wilks calls "lexical preference" information [Wilks 1985]. This would seem to be a combination

[2] This is a preference for "minimal attachment", which prefers analyses which create fewest nodes in a parse tree. This is often taken to have precedence over a preference for "right-attachment", which prefers analyses in which we attach to the current constituent under construction. Pereira [1985] shows that in a shift-reduce parser right-attachment corresponds to preferring shifts over reductions, and minimal attachment corresponds to preferring to reduce as much as possible.

of verb subcategorisation information (how many arguments a verb must have) and selection restrictions or preferences on the types of objects that can fill these arguments. For example, one "positions" objects (such as dresses) on things (such as racks), i.e. "positions" is happy with two arguments to its right, one the object being positioned and one the place where the object is being positioned. Hence in (5) we would prefer VP-attachment (5c) since this would preserve lexical preferences. On the other hand, for (4), the lexical entry for "want" would say that it is happy with a single argument to its right describing the object desired. This would make the system attempt to interpret as much text after the verb as possible as a single NP argument for "want". This would seem to correctly prefer (4b), where the PP is attached to the NP to give us a single object "the dress on the rack". But we can find a counter-example to this:

(6) "I want that report on my desk by Monday morning!"

where only "that report" should be interpreted as the object desired.

There is a more specific disambiguation strategy available but it can only be used for *definite, singular NPs used referentially.* This strategy was probably first used by Winograd's SHRDLU [Winograd 1972]. Winograd's approach has been followed up in empirical work by Crain & Steedman [1985] and Altmann [1987], and has been used in Hirst's Absity, Pollack & Pereira's CANDIDE, Mellish's system, Haddock's system and T42. Winograd's reasoning on cases such as example (2) is that you can tell whether the phrase "in the box" is part of the noun phrase "the block in the box" or part of the noun phrase "the box on the table" by referring to the context of utterance. If the expression "the block" produces a unique referent, then there would be no need to give any further information: analysis (2c) would be adopted. If, however, a unique block is not identifiable from "the block" alone, we would treat the PP as extra information to help us to identify which block is being talked about: analysis (2b) would tell us to expect to find a unique block in a box. Note crucially that this depends on the noun phrases being singular, definite and used referentially, as it is only with these noun phrases that we would expect to find a unique referent in memory already[3].

Crain & Steedman [1985] performed psycholinguistic experiments that investigated the role of contextual knowledge in human processing of "garden path" sentences. For example:

(7) "The merchant sold the wheat for a lot of money wanted to kill Launcelot."

is a sentence which, presented to you, as it is, without any real context, may cause you mistakenly first to take "sold" as the verb of the sentence, when in fact "sold" here is a past participle introducing a relative clause (paraphrased as "The merchant *who was* sold the wheat for a lot of money...") and "wanted to kill Launcelot" is the verb phrase. Crain & Steedman argue

[3] Chapter 5 discusses what it means for a NP to be definite, singular and used referentially, and for a "referent to be in memory".

that it is not sentence structure which induces garden pathing but rather it is certain contexts, particularly the 'null' context. In ordinary discourse, restrictive relative clauses such as this would only be used if the speaker believed that the hearer needed more information to be able to identify which of several merchants was being talked about. The hearer, on hearing the phrase "The merchant", would notice that this did not identify which of the several potential referents was being singled out and so, without garden-pathing, would realise that "sold" introduces a relative clause that would help him to work this out. So for example, with (7) given a bit more context, you should no longer garden path:

(8) "*In the market there were two merchants. One of them had bought some wheat for a lot of money, while the other had bought some very cheaply. The merchant sold the wheat for a lot of money wanted to kill Launcelot.*"

Crain & Steedman formulated the results of their experiments into a set of principles. Their most specific principle, and the one of interest to us, is:

> *The Principle of Referential Success*: If there is a reading that succeeds in referring to an entity already established in the hearer's mental model of the domain of discourse, then it is favoured over one that does not.[4]

Hirst in his Absity system [1987] (see Chapter 2) used this principle to suggest when prepositional phrases should be attached to definite NPs. For each reading of a sentence, his "Semantic Enquiry Desk" interrogates the knowledge base to determine whether that reading is referentially successful. An analysis which is referentially successful is preferred over one which is not[5]. The parallel to Winograd's system is obvious. Hirst is saying that for (2b) to be referentially successful there must be a block in a box in context, while for (2c) to be referentially successful there must be a box on a table in context: the preferred reading is the one which is referentially successful. The idea of choosing a reading that is referentially successful is a more specific form of saying one chooses the reading whose presuppositions are satisfied.

Altmann [1987] reassesses Crain & Steedman's work in the light of further experiments. He notes a problem with the Principle of Referential Success: a hearer can only choose between alternative analyses once he has identified the intended referent. For example, for Hirst and for Crain & Steedman the decision that "raced" is a past participle in the last sentence of (9) and not a tensed verb would have to await the end of processing the underlined text:

(9) "*There were two horses. One was raced past the barn, while the other was raced past the stable. The horse raced past the barn fell.*"

[4] Their more general principles, which subsume this one, require presupposition satisfaction and *a priori* plausibility.

[5] I previously said that this approach is limited to singular, definite noun phrases used referentially. Crain & Steedman and Hirst, in also using some broader principles (see previous footnote), claim to extend the applicability of the approach.

Clearly, the decision can, in fact, be made at the end of processing "The horse", at which point we know that we have not identified a unique horse: further information is expected.

Altmann puts this idea into a principle based on referential failure rather than on referential success:

> *The Principle of Referential Failure:* If a referring expression fails to refer to an entity already established in the hearer's mental model of the domain of discourse, then an analysis that treats subsequent material as a modifier for that referring expression (i.e., as providing information that may lead to successful reference), will be favoured over one that does not.

This implies looking for referents as soon as a noun phrase can be syntactically closed.

This is the idea used in Haddock's system and in T42. These systems try to close noun phrases as soon as possible. If a noun phrase expects a unique referent to have been found (e.g. if it is singular, definite and used referentially) and a unique referent has *not* been found (as signalled by the constraint satisfier), then closure of the NP will fail. To close the NP at this point would be premature. Some modifying phrase to refine the NP description such as a PP or a restrictive relative clause is needed to help identify a unique referent.

I will emphasise that this approach works in an obvious way only for cases of singular, definite reference. For a fuller solution we must make use of other sources of knowledge as well, e.g. syntactic preference, verb subcategorisation and lexical preferences, and general world knowledge. However, for singular, definite reference the Referential Success/Referential Failure approach does seem to work all the time[6]. For the fuller problem, Hirst [1987] describes a system which attempts a solution for all cases of PP attachment. Each of the strategies that I have mentioned chooses its preferred reading and then a decision routine chooses amongst their recommendations. Basically, faced with conflicting choices it takes the referential success strategy as the most important, then the lexical preferences strategy and uses syntactic preferences as a last resort. With the use of several different types of knowledge, Hirst manages to get a very good coverage of examples.

The next section gives a worked example to show how T42 makes use of referential failure to resolve PP attachment decisions.

[6] Hirst's putative counter-examples involve "non-restrictive PPs":

"Romeo thinks of Juliet with nothing on." [after Hirst 1987, p.165]

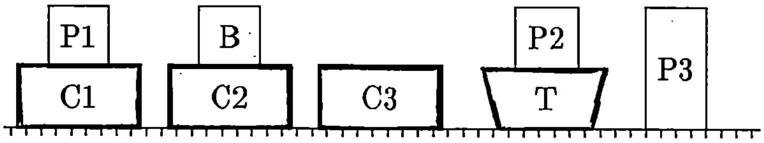
where, since we are able, within the context of the Shakespearean play, to identify a unique Juliet, the Referential Success approach would attach the PP to the verb: Romeo strips off before he does his dreaming about Juliet. The more likely reading however goes against this: Romeo is imagining Juliet's naked body. It is possible that this is not a referential use of the NP "Juliet" and so is no counter-example at all.

4.2 A PP Attachment Example

The example presented here should serve two purposes: first, it should make Chapter 3's description of T42 clearer, and secondly it should illustrate definite NP processing more fully. The example follows Haddock in that it involves only a noun phrase and a prepositional phrase which might or might not be attached to the noun phrase. Haddock does not describe the issues of attaching to a verb phrase: we are to assume that if the system as described here does not recommend attachment to the noun phrase then by default the PP will be attached to something else (e.g. the main verb).

Note that the noun phrase is definite and singular, and is to be taken as used referentially. Thus the approach to PP attachment problems being described here is based on the notion of referential failure described in the previous section. I reiterate that referential failure furnishes only a partial solution to PP attachment problems because it is restricted to deciding whether to attach to definite NPs. In the general case, other knowledge must be brought to bear if we are to be able to make an attachment.

Consider the following scenario in which there are three caskets (C1, C2 and C3), three portraits (P1, P2 and P3), a book (B) and a tub (T):



The discourse model (more specifically, the non-immediate context) contains the following clauses representing this scenario (these clauses are in fact set up before discourse processing begins but they could just as well arise from previous text):

```
PORTRAIT(P1)        PORTRAIT(P2)        PORTRAIT(P3)
CASKET(C1)          CASKET(C2)          CASKET(C3)
BOOK(B)             TUB(T)
IN(P1, C1)          IN(B, C2)           IN(P2, T)
```

Thus we have a discourse model containing information about the entities P1, P2, P3, C1, C2, C3, B and T.

The noun phrase I shall be processing is "the portrait in the casket". In this context, as can be seen from the picture, if the PP is attached to the NP, this uniquely identifies P1 as its referent. What we are interested in is how the system decides that this is the right attachment to make. Before working through the example, I wish to prefigure some of its important features:

1. The example will show how feedback from the constraint satisfier tells the parser to attach the prepositional phrase "in the casket" to the noun phrase "the portrait". Since more than one referent for "the portrait" as a noun phrase in its own right will be found, the uniqueness

requirement on singular, definite NPs will not be satisfied, and thus T42 will know that it has closed the noun phrase prematurely. More information is expected to help it to identify a unique referent. This information might come in the form of a restrictive relative clause or, as in fact it does, as a prepositional phrase. Once the noun phrase is modified by the prepositional phrase, giving a more restrictive description, a unique referent will have been identified. This processing embodies the idea of referential failure.

2. In the phrase "the portrait in the casket", there are two definite NPs. T42 will not demand a unique portrait and a unique casket. Rather it will demand a unique portrait in a casket, and a unique casket with a portrait in it. It is particularly important to note that the casket does not have to be unique except in the context of caskets with portraits in them. A number of systems, possibly including Hirst's Absity, incorrectly place this constraint on the inner NP due to a too simplistic compositional and semantic treatment of uniqueness. Here uniqueness is treated as non-compositional and pragmatic. This is described more fully in Section 4.3.

3. The system regularly consults the discourse model to allow findings about discourse model entities to guide its processing. This means that when T42 comes across the word "in", for example, even though it is only part way through the utterance, it immediately looks up all the instances of the IN-relation in the discourse model: it immediately starts to consider all objects that are known to be *in* other objects, and all those objects that have things *in* them. As a consequence, we get a representation of intermediate phrases. For example, having processed "the portrait" (step 4 below), T42 will have the referents for "the portrait", P1, P2 and P3, showing that it is 'thinking of' three possible portraits that could have been intended at that point in the sentence. At the end of processing "the portrait in" (step 6), it will have reduced these referents to only two portraits, P1 and P2: those that are in things. By the end of the input phrase (step 10), it will have reduced this to P1, the portrait in the casket.

The relevant lexical entries for the example are as follows:

word	*category*	*meaning*
"casket"	Nx	CASKET(x)
"in"	$(\mathrm{N}\backslash\mathrm{N}x_0)/\mathrm{NP}x_1$	IN(x_0, x_1)
"portrait"	Nx	PORTRAIT(x)
"the"	NPx/Nx	$\star$UNIQUE(x)

Note that "in" is, in fact, categorially ambiguous in categorial grammar. In the form shown here, it heads PPs that can be attached to NPs; in one of its other forms it heads PPs that can be attached to VPs. I have left these other forms out to keep the example simple. If they had been included, I would have needed to show the parser backtracking to try each alternative out. This omission is not important: the main point of the example is to show how information about discourse model entities can force the parser to leave the NP "the portrait" open for attachment of some modifier to help identify a unique referent. Note further that I am not claiming that such a simplistic lexical semantics for "in" is adequate: it just keeps the example simple.

In the exposition, I will show the stack used for parsing, the remaining input and the contents of the immediate linguistic context (ILC), i.e. the variables, their satisfaction sets and the constraints. We start with all structures empty except for the input buffer, which contains the whole phrase:

Stack: $\lfloor\rfloor$ Input: "the portrait in the casket"

Variables: None Constraints: None

1. Read in "the". Its category is put onto the stack; its meaning entry is sent to the constraint satisfier (CS). The $\star$ on its UNIQUEness constraint tells us to do nothing yet. (It is a cardinality constraint which will not be tested until syntactic closure of the noun phrase – see steps 3 and 10). Notice that variables get renamed on shifting:

Stack: $\lfloor \mathrm{NP}x_0/\mathrm{N}x_0 \rfloor$ Input: "portrait in the casket"

Variables: None Constraints:
$\star$UNIQUE(x_0)

2. Read in "portrait"; its category is Nx_1 and its semantics is PORTRAIT(x_1). The latter is sent to the CS. Variable x_1 will have a satisfaction set of P1, P2 and P3, i.e. all discourse model entities which are portraits (this is determined by querying the discourse model):

Stack: Nx_1
NPx_0/Nx_0 Input: "in the casket"

Variables: Constraints:
x_1 = {P1, P2, P3} $\star$UNIQUE(x_0) PORTRAIT(x_1)

3. Reduce NPx_0/Nx_0 and Nx_1 to NPx_0 using forward application. Recall that our parsing strategy prefers to do reductions before all else and this is why this is chosen. This reduction creates a constraint EQUAL(x_1, x_0) due to 'equating' Nx_0 and Nx_1 in the reduction. This tells us that the satisfaction sets of x_0 and x_1 are equal.

There is an alternative to reducing at this point and that is to type-raise (shifting is inhibited because there are reductions and type-raisings). Categories of type N may type-raise to N/(N\N) to allow prepositional phrase attachment. Since, as has been mentioned, this is dispreferred to the reduction, it is put onto the agenda as a parse that might be returned to later.

The effect of these two changes is:

Stack: $\boxed{\mathrm{NP}x_0}$ Input: "in the casket"

Variables:			Constraints:
x_0, x_1	$=$	`{P1, P2, P3}`	⋆`UNIQUE`(x_0) `PORTRAIT`(x_1) `EQUAL`(x_1, x_0)

Note that the agenda is not shown.

Things do not stop here: the top of the stack now shows that a NP, "the portrait", has been recognised. This is where any constraints with ⋆s are woken up, i.e. syntactic closure of NPs activates the cardinality constraints set up by use of the word "the". In this case, we have a `UNIQUE`ness constraint on x_0. The satisfaction set of x_0 is tested to see if it contains only a single item. It does not. The closure of this NP has been premature. We need more information (such as that which might come from a PP) to help us to identify a unique referent. Thus, this parse is discarded and we reactivate the parse on the agenda. After restoring the item from the agenda, we have:

Stack: $\boxed{\begin{array}{l} \mathrm{N}x_1/(\mathrm{N}\backslash\mathrm{N}x_1) \\ \mathrm{NP}x_0/\mathrm{N}x_0 \end{array}}$ Input: "in the casket"

Variables:			Constraints:
x_1	$=$	`{P1, P2, P3}`	⋆`UNIQUE`(x_0) `PORTRAIT`(x_1)

4. The top two items of the stack can be reduced. (Again, the option to shift is inhibited). The reduction generates the equality constraint `EQUAL`(x_1, x_0):

Stack: $\boxed{\mathrm{NP}x_0/(\mathrm{N}\backslash\mathrm{N}x_1)}$ Input: "in the casket"

Variables:			Constraints:
x_0, x_1	$=$	`{P1, P2, P3}`	⋆`UNIQUE`(x_0) `PORTRAIT`(x_1) `EQUAL`(x_1, x_0)

5. Read in "in". x_2 and x_3 are introduced into the Constraint Satisfaction Problem (CSP) and we search for instances of the `IN` relation in the discourse model:

Stack: $\boxed{\begin{array}{l} (\mathrm{N}\backslash\mathrm{N}x_2)/\mathrm{NP}x_3 \\ \mathrm{NP}x_0/(\mathrm{N}\backslash\mathrm{N}x_1) \end{array}}$ Input: "the casket"

Variables:			Constraints:
x_0, x_1	$=$	`{P1, P2, P3}`	⋆`UNIQUE`(x_0) `PORTRAIT`(x_1) `EQUAL`(x_1, x_0)
x_2	$=$	`{P1, B, P2}`	`IN`(x_2, x_3)
x_3	$=$	`{C1, C2, T}`	

It is important to note that <P1, C1>, <B, C2> and <P2, T> were found as *pairs* satisfying IN(x_2, x_3). This information is not lost by the CS, although it is not shown in the diagrams. In other words, only these three pairs satisfy the constraints; other members of the Cartesian product of x_2 and x_3 are not legal. Furthermore, if some constraint were to rule C1 out of x_3, say, then this would have the effect of ruling P1 out of x_2 too, i.e. the deletion would be propagated to any mutually dependent constants.

6. A reduction of the top two stack items is possible and, as ever, is preferred to any other options. This reduction (to NPx_0/NPx_3) generates the constraint EQUAL(x_2, x_1). This says that the thing that is IN something (x_2) is equal to the portrait (x_1). The satisfaction sets of these are equated: this rules out P3 from x_1 since it is a portrait that is not in anything. It also rules B out of x_2 since it is not a portrait. In removing B from x_2, we recall that there is a dependency between it and C2 in x_3 (i.e. C2 is the thing that B is in); we therefore remove C2 from x_2 also:

Stack: $\boxed{\mathrm{NP}x_0/\mathrm{NP}x_3}$ Input: "the casket"

Variables:			Constraints:		
x_0, x_1, x_2	=	{P1, P2}	⋆UNIQUE(x_0)	PORTRAIT(x_1)	EQUAL(x_1, x_0)
x_3	=	{C1, T}	IN(x_2, x_3)	EQUAL(x_2, x_1)	

Before proceeding, it is worth taking stock of what we have so far. We have read in the phrase "the portrait in" and the stack shows that we have category NP/NP. This seems reasonable; it says that we are waiting for a NP in order to 'become' a NP which is the sort of constituent "the portrait in" seems to be. This kind of 'intermediate category' is hard to get in conventional phrase-structure grammars. Furthermore, x_0 shows us that, at this stage, we are 'thinking about' portraits that are in things (namely, P1 and P2 but not P3) and x_3 shows us that we are 'thinking about' things which have portraits in them (namely, C1 and T). So we have a characterisation of the possible referents so far.

We can also see that the ILC (variables and constraints) simply captures some simple formulas of clausal form logic, albeit in a form amenable to the constraint satisfier. Specifically, it should be clear that we can read the following two readings out of the ILC[7]: PORTRAIT(P1) ∧ IN(P1, C1) and PORTRAIT(P2) ∧ IN(P2, T). This emphasises the point I was making in Chapter 3 that the ILC and the NIC contain information with the same interpretations, but use different data structures.

7. Read in "the". Its meaning has a ⋆, so we do nothing else yet:

[7] Ignoring the uniqueness constraint and showing them as first-order predicate calculus rather than in clausal form.

Stack: $\boxed{\begin{array}{l} \mathrm{NP}x_4/\mathrm{N}x_4 \\ \mathrm{NP}x_0/\mathrm{NP}x_3 \end{array}}$ Input: “casket”

Variables:			Constraints:		
x_0, x_1, x_2	=	{P1, P2}	$\star$UNIQUE(x_0)	PORTRAIT(x_1)	EQUAL(x_1, x_0)
x_3	=	{C1, T}	IN(x_2, x_3)	EQUAL(x_2, x_1)	$\star$UNIQUE(x_4)

8. A reduction is possible and it generates the constraint EQUAL(x_4, x_3):

Stack: $\boxed{\mathrm{NP}x_0/\mathrm{N}x_4}$ Input: “casket”

Variables:			Constraints:		
x_0, x_1, x_2	=	{P1, P2}	$\star$UNIQUE(x_0)	PORTRAIT(x_1)	EQUAL(x_1, x_0)
x_3, x_4	=	{C1, T}	IN(x_2, x_3)	EQUAL(x_2, x_1)	$\star$UNIQUE(x_4)
			EQUAL(x_4, x_3)		

9. Read in “casket”. Its semantics, CASKET(x_5), adds x_5 to the CSP with a satisfaction set of all caskets, i.e. C1, C2 and C3:

Stack: $\boxed{\begin{array}{l} \mathrm{N}x_5 \\ \mathrm{NP}x_0/\mathrm{N}x_4 \end{array}}$ Input: None

Variables:			Constraints:		
x_0, x_1, x_2	=	{P1, P2}	$\star$UNIQUE(x_0)	PORTRAIT(x_1)	EQUAL(x_1, x_0)
x_3, x_4	=	{C1, T}	IN(x_2, x_3)	EQUAL(x_2, x_1)	$\star$UNIQUE(x_4)
x_5	=	{C1, C2, C3}	EQUAL(x_4, x_3)	CASKET(x_5)	

10. There are now two options. The dispreferred option is to type-raise the stack top, Nx_5, to Nx_5/(N\N$x_5$): this option goes onto the agenda.

The preferred alternative is to reduce the stack contents using forward application to give NPx_0 and so generating the constraint EQUAL(x_5, x_4). This constraint says that the caskets that we wish to continue to consider (x_5) must be equal to the things which contain portraits (x_4), i.e. we are only interested in caskets with portraits in them. The effect of the constraint is to rule C2 and C3 out of x_5, and to rule T out of x_4 (T is a tub not a casket, C2 contains a book not a portrait, and C3 also does not contain a portrait since it is empty). When we rule T out of x_4, we must also remove P2 from x_2 because these two objects are inter-dependent:

Stack: $\boxed{\mathrm{NP}x_0}$ Input: None

Variables:			Constraints:		
x_0, x_1, x_2	=	{P1}	⋆UNIQUE(x_0)	PORTRAIT(x_1)	EQUAL(x_1, x_0)
x_3, x_4, x_5	=	{C1}	IN(x_2, x_3)	EQUAL(x_2, x_1)	⋆UNIQUE(x_4)
			EQUAL(x_4, x_3)	CASKET(x_5)	EQUAL(x_5, x_4)

The stack top shows that we have syntactically closed a NP. Hence, the uniqueness constraints are now invoked. There are two of them: one on x_0 and one on x_4. Both are satisfied since both these satisfaction sets contain only one item. Thus this analysis is given the 'thumbs-up': all of the input has been read in, a NP has been recognised and the referent of the NP is P1 as we expected. The option that remains on the agenda could now be reactivated if we wanted to check for all parses. It will almost immediately fail since there is no more input to shift.

This concludes the example. It should be clear that referential failure was used to keep the NP open at step 3 and that T42 not only keeps syntactic information about what input it has processed so far but also knows about the referents of the input so far. The example also showed how the components of the system interact. The shift-reduce parser feeds constraint-based meaning from the lexical entries and from the effects of the parsing (specifically, the EQUAL and noun phrase syntactic closure constraints) to the constraint satisfier. The constraint satisfier assesses the consistency of these constraints, calling upon the inference engine to supply facts from the discourse model about possible referents.

For the example, the constraint satisfier will have accumulated the following clauses:

⋆UNIQUE(x_0)	PORTRAIT(x_1)	EQUAL(x_1, x_0)	IN(x_2, x_3)	EQUAL(x_2, x_1)
⋆UNIQUE(x_4)	EQUAL(x_4, x_3)	CASKET(x_5)	EQUAL(x_5, x_4)	

It is possible to get a simple logical form as the representation of the example input phrase. We can ignore the UNIQUEs which I discuss in the next section. We can rename EQUAL variables and then discard the EQUALs constraints. Then if we conjoin the remaining constraints and existentially quantify the variables, we obtain:

$$\exists x_0 \exists x_4\ (\mathtt{PORTRAIT}(x_0) \wedge \mathtt{IN}(x_0, x_4) \wedge \mathtt{CASKET}(x_4))$$

This seems reasonable as a Russellian-style analysis of sentence meaning, though strictly, if we are to give a Russellian analysis, we would have to treat the UNIQUEness constraints semantically too. I will show in the next section that this is not necessarily desirable: if we are not careful it will give a logical form that would demand both a unique portrait and a unique casket, rather than a unique portrait in a casket.

But perhaps more important than having obtained the logical form just given for the example NP is having obtained its referent, P1, since this constant can be used in other parts of the logical form (e.g. as the argument of some other predicate).

4.3 The Uniqueness Constraint

According to Russell [1905 (1975)], the uniqueness constraint on definite noun phrases is semantic and can be handled in a first-order logic. He would give the noun phrase (10a) the logical form (10b):

(10) a. "the portrait"
b. $\lambda P\ (\exists x\ (\texttt{PORTRAIT}(x) \wedge \neg \exists y\ (\texttt{PORTRAIT}(y) \wedge (x \neq y)) \wedge P(x)))$

The uniqueness constraint is entailed by the logical form. Many NLP systems give this analysis too and go one step further: not only is uniqueness semantic (and hence entailed), it is also part of compositional meaning. Haddock's work on PP attachment suggests that one or more facets of this traditional analysis must be abandoned. For example, one option (which [Haddock 1987a, 1987b] do not discuss) is that the uniqueness constraint could still be compositional and semantic but would be given a higher-order translation relating an individual and a proposition.[8] The option which Haddock takes is to lay the blame on compositionality: his uniqueness constraint behaves in a non-compositional manner. I have adopted this idea and also argue below that uniqueness is pragmatic. Either way, this is why I have labelled the constraint with a $\star$ and have talked of it as being 'special'. Given its difference in status, it is not unreasonable that it should be handled differently from other constraints. Specifically, it is pended until syntactic closure of the constituent it applies to. Then it is woken up and used to test whether a unique referent has been found. The uniqueness is absolute: there must be only one referent in the satisfaction set under consideration. It is up to other parts of the system to ensure that only referents that are "salient" or are "in focus" were considered.

First, I will explain why I believe uniqueness is pragmatic. In this I follow Kempson [1975, pp.109-111]. She says that if Russell is right the proposition expressed by (11a) would entail (11b):

(11) a. "The portrait has been touched up."
b. 'There is one and only one portrait.'

But if (11a) entails (11b), then (11a) should be false if (11b) (or any of its other entailments) is false. In other words, a sufficient condition for me to be able to deny (11a) should be the falsity of (11b). Yet (12a), which expresses this, seems incoherent. Indeed it seems incoherent in the same way that (12b) does (where (12b) has an indefinite, which will not have a uniqueness constraint, instead of a definite):

[8] This was pointed out to me by Steve Pulman.

(12) a. ??"It is not true that the portrait has been touched up because there is more than one portrait."
b. ??"It is not true that a portrait has been touched up because there is more than one portrait."

Rather it seems that the conditions that make an utterance with a definite NP true are the same as those that make an utterance with an indefinite NP true, i.e. there is no *semantic* distinction between (13a) and (13b):

(13) a. "The portrait has been touched up."
b. "A portrait has been touched up."

However, (13a) has an additional *pragmatic* implication of uniqueness. The *truth* of (13a) is not affected by whether the hearer can identify a *unique* referent; its *appropriateness* is.

Ultimately Kempson (if I have understood her argument) attributes the uniqueness constraint on definite NPs to generalised *conversational* implicatures. I am not sure I agree: I think uniqueness is a *conventional* implicature. It has all the properties of conventional implicatures (see Chapter 1): it seems to be conventional, non-calculable, and not cancellable in positive utterances (??"The portrait has been touched up and there may be more than one portrait."); it survives in negative utterances ("The portrait wasn't touched up."), is detachable (by using the indefinite article), and is non-reinforceable (??"The portrait has been touched up and there is only one portrait."). However, deciding whether something is a conventional implicature or a (very) generalised conversational implicature is notoriously difficult. Fortunately for the purposes of this thesis, this distinction will not matter. All that is important is that I take uniqueness to be pragmatic and therefore it justifiably receives a different treatment from semantic parts of meaning.

Secondly I will repeat Haddock's argument for why the uniqueness constraint should be noncompositional. If it were compositional, a problem would arise with definite NPs embedded within other definite NPs, such as "the portrait in the casket".[9] The compositional semantics approach would produce the following logical form for this:

$$\lambda P\ (\exists w \exists y\ (\texttt{PORTRAIT}(w) \wedge \neg\exists x\ (\texttt{PORTRAIT}(x) \wedge (w \neq x)) \wedge \texttt{IN}(w, y) \\ \wedge\ \texttt{CASKET}(y) \wedge \neg\exists z\ (\texttt{CASKET}(z) \wedge (y \neq z)) \wedge P(x))$$

Now this seems plainly wrong: it demands a unique portrait and a unique casket. What it does not make clear is that what is actually wanted from this context is *a unique portrait in a casket* and *a unique casket with a portrait in it.* This simple logical form does not show that the search for a unique casket should be constrained by the portraits that are in things. Haddock [1987a, 1987b] recognises this and claims it as one of the big problems that his

[9] This problem does not necessarily arise if uniqueness is semantic and compositional but higher-order.

system and hence T42 overcome[10]. T42 will place uniqueness requirements only on caskets with portraits in them and portraits that are in caskets, and not portraits or caskets on their own.

There is another problem with uniqueness constraints on definite NPs which I have not mentioned so far. The first point to make is that we should hardly ever actually retrieve *all* the entities that satisfy a constraint from the discourse model. Rather, we should only retrieve those entities that are "salient" or "in focus" in some way. A good review of various notions of focussing is [Carter 1987]. Focussing approaches to this problem are very much heuristical. They sometimes fail to find an entity or sometimes find the wrong entity. Despite these problems, if one is to handle texts of any length one must introduce such mechanisms. T42 has no focussing mechanism at present; it has only been tested on short (artificial) texts (no more than 3 sentences long) which have been specially constructed to circumvent the need for focussing.

Unfortunately, focussing alone may not be enough. There are cases where a unique referent cannot be distinguished even from the salient ones until further information is processed and where this further information comes arbitrarily late in the sentence (well beyond syntactic closure of the NP). (This point is made by Ritchie in his critique of interleaved processing [Ritchie 1983] — see Chapter 2). For example:

(14) a. "*Caliban bought a book from Prospero. He gave him* $\left\{\begin{array}{l}\textit{too much money."}\\ \textit{too little change."}\end{array}\right\}$

b. "*The Prince of Wales became king in 1413.*"

In (14a) the underlined pronouns can refer to either Caliban or Prospero. You cannot decide which until you get to the end of the sentence when real-world knowledge resolves this for you. In (14b), of all the people who have held the title "Prince of Wales", you can only pick out the correct one once you get to the date at the end of the sentence.[11] Presently T42 would find these infelicitous as the uniqueness constraint cannot be satisfied at NP closure. My attempts to relax the uniqueness constraint have merely admitted more readings than seems reasonable. A future enhancement might be to allow a delayed decision (the CANDIDE system might be able to do this). However,

[10] Haddock says that even other incremental systems such as Hirst's Absity [Hirst 1987] get this wrong too. Whether Haddock is right in this criticism is not clear. Hirst certainly seems to recognise the problem. He says: "... *the ocelot with the blue chipmunk* presupposes that there is just one such ocelot available for reference ...; the plausibility and existence of an **ocelot** and a **blue chipmunk** continue to be presupposed, but their uniqueness is no longer required." [*ibid.*, p.170]. However, his semantic form for the similar NP "the man with the telescope" is

```
(the ?x (man ?x (attr=(the ?y (telescope ?y)))))
```

This is used as a query against his knowledge base. **the** is a query command that searches for a unique referent. And so a unique telescope seems to be demanded. In support of Haddock's criticism, there appears to be nothing in the semantics of Hirst's query language that would remove the uniqueness constraint from the inner NP.

[11] It is because we have to wait until the end of the sentence to resolve these that Ritchie questions the value of interleaved processing.

before such enhancements are made, I believe this kind of data needs more attention, with particular regard to the contexts in which such utterances might arise. For example, I do not believe (14b) is something that can be uttered 'out of the blue', and inasmuch as it would be uttered in some context where, e.g., a particular period in history was under discussion, this might help to give a strong preference for the referent of the NP early on in the utterance.

Finally in connection with the uniqueness constraint, I must make some mention of *plural* definite noun phrases and consider how easy (or otherwise) it would be to extend T42's processing mechanisms to handle these. (It must be pointed out that to some extent this is little more than a 'hand-wave' since I have not given plurals much thought at all). The most obvious extension would be to say that the members of a satisfaction set would no longer be atomic constants but could be sets of constants. For example, "portraits" in the definite NP "the portraits" would be a word that would introduce into the CSP a variable whose extension might be as follows: {{`P1, P2`}, {`P1, P3`}, {`P2, P3`}, {`P1, P2, P3`}}, i.e. all non-empty, non-singleton members of the power-set of the portraits in the discourse model. Notice that it does not follow that the only referent we would get would be the 'maximal' referent {`P1, P2, P3`}. Rather, a set of candidate referents is retrieved. Some of these candidates might then be ruled out by virtue of being of low salience. If this still leaves a choice, the uniqueness constraint would not be satisfied so T42 would expect some modifying phrase. For example, if the 4 candidates given above were equally salient, T42 might leave the NP open and try to process a modifying phrase. If the full NP turns out to be "the portraits that are in things" then this would uniquely identify the set {`P1, P2`}. Obviously this 'proposal' requires much more work.

::::

In this chapter I have given an overview of how to make PP attachment decisions. I have concentrated on the use of referential failure for making decisions about attachment to singular, definite NPs used referentially. To consolidate Chapter 3's description of T42, I gave a simple PP attachment example in some detail. This also demonstrated an advantage of incremental systems: the attachment decision was made as soon as possible and without the computation of unneeded structures. In the next chapter I will discuss definite NPs more fully and relate T42's processing of them to the work of Russell and Strawson. This will prepare the ground for a discussion of extending T42 to other parts of speech and to handling presuppositions in a way that is uniform with definite NP processing.

Chapter 5

Definite Noun Phrases in T42

This chapter considers the processing of definite NPs in T42 from a more theoretical perspective. Up to now, I have given examples but I have not situated these examples in any real theory. It is the purpose of this chapter to give this theory and draw out some of the assumptions I have made. This will enable me to relate definite NP processing to presupposition processing.

Section 5.1 gives a brief explanation of what is meant by the phrase "definite noun phrase" in this thesis. Section 5.2 distinguishes three possible uses of definite NPs: referential, attributive and generic. It indicates that T42 is primarily concerned with referential uses. In Section 5.3, I point out that T42 can in fact be run in one of two 'modes': in the one that has been discussed in previous chapters, entities are retrieved from a discourse model and used to solve a Constraint Satisfaction Problem (CSP); in the other mode, constraints are simply accumulated and the discourse model is not consulted. I relate these two modes to Strawson's and Russell's theories of definite NPs and to referential and attributive uses of definite NPs.

5.1 Definite Noun Phrases

My characterisation of definite NPs is syntactic: they are to be recognised by their form. I make this point explicit because I think that sometimes the terms "definite *descriptions*" and even "definite NPs" are used in a way that is not just syntactic but also implies a certain use of the NP (usually the referential use — see next section). For example, while for me pleonastic "it" is a definite NP, some might not choose to describe it as a definite description, precisely because it does not refer (as in "*It seems that 'it' need not refer.*").

A definite NP for me would be one of the following forms:

- proper nouns, e.g. "Olivia", "Sir Toby Belch"
- certain pronouns, e.g. "I", "you", "he", "she", "it", but *not* "who", "someone"
- common nouns preceded by a definite article (and optional modifiers), e.g. "the dog", "the brown dog"

- common nouns preceded by a possessive pronoun, e.g. "his dog"
- common nouns preceded by a possessive definite NP, e.g. "Launce's dog", "the boy's dog"

These may all be followed by PPs or relative clauses (although proper names and pronouns rarely are). There are other examples not really considered in this thesis, e.g. demonstrative pronouns ("this", "that"), common nouns preceded by demonstrative adjectives ("this dog") and cases involving certain quantifiers (e.g. "all of the dogs").

I should also make it clear that I am looking only at *singular, count* nouns. So, while "we", "the dogs", "their dogs", "those", "these dogs", bare plurals as in "Dogs bark" and mass terms as in "Milk is nice" are strictly definite NPs, they are not considered herein.[1]

5.2 Uses of Definite Noun Phrases

The definite NP examples given in the previous two chapters were processed with the assumption that T42 should look for an entity in the discourse model. This corresponds in an obvious way to only one use of definite NPs, the referential use. Three main uses of definite NPs are, in fact, recognised: referential, attributive and generic. It is hard to *define* any of these. The conditions for a noun phrase being used referentially, for example, are not agreed upon. What is agreed is that you cannot, in general, tell the intended use from the syntactic form. There are some syntactic constructions which 'favour' one reading over another but these are not clear-cut (see [Fawcett 1986] for more details). In general, one must rely on the context of utterance to determine which use is intended. The relevant points about these uses are as follows.

- **Referential Uses**

The referential use is the main case considered in this thesis. In this use the speaker intends to single out a particular object for further discussion: the speaker 'has a particular object in mind' and uses a description which she believes will enable the hearer to identify the same object. The description need not be correct: it must simply be enough to enable identification of the referent (e.g. if I say "the present King of France", you might, in the right circumstances, still understand me to be referring to President Mitterand; see [Clark & Marshall 1981] for more details on what "the right circumstances" might be). This is the case we have looked at in previous chapters and will be explored further in this thesis.

One of the issues that must be considered is what kind of objects get referred to in referential uses. In this thesis the objects need only be entities in the discourse model, which need not denote simple real world objects. For example, cases where pronouns behave like bound variables (as in "*Every man who owns a donkey beats it.*") are reconcilable with a referential treatment

[1] But for a brief discussion of plurals, see the end of Section 4.3 in the previous chapter.

if they are allowed to pick up discourse model entities whose denotations are not simple real world objects (as in Discourse Representation Theory [Kamp 1984]).

A referential use corresponds to an "anaphoric" use if the term "anaphor" is used in its broad NLP sense. For linguists, anaphora refer to textual antecedents. In NLP, anaphora pick up objects that have been introduced by previous text, are evoked by the physical discourse setting ("exophora") or are inferable from the setting or previous text; in short, the objects picked up by anaphora are entities in the discourse model, irrespective of how they came to be there. And, since parts of my discourse model may contain entities which behave like discourse referents in Discourse Representation Theory (DRT) and so do not denote simple objects (see Chapter 9), I am using the term anaphor perhaps even more liberally than it has typically been used in NLP.

- **Attributive Uses**

The attributive-referential distinction was introduced by Donnellan [1966]. It is generally taken to be a pragmatic ambiguity (I discuss what a pragmatic ambiguity might be, with reference to [Stalnaker 1972], in Chapter 8). In attributive uses of definite NPs the speaker does not have a particular object in mind. She cannot use the description inaccurately. Her description is essential: her utterance predicates something about whoever or whatever fits the description. The attributive use of definite NPs is mentioned again in Chapter 9 in connection with definite NPs that do not have their usual presuppositions, hence it is worth spending more time characterising them properly now.

Donnellan's main example concerns Inspector Dogberry, say, coming onto the scene of what he takes to be a crime: Duncan's body, horribly mutilated and freshly deceased lying on the floor. Suppose Dogberry utters (1a) below, i.e. "*<u>Duncan's murderer</u> is insane*". If he intends that the description "Duncan's murderer" should pick out an entity, Macbeth say, for the hearer, then this is a referential use. If however Dogberry does not intend such a referent to be picked out, he uses the definite NP attributively to mean 'whoever murdered Duncan' (i.e. whoever could commit such a ghastly crime) is (or must be) insane.[2] Another example is (1b), where the NP "*The strongest man in the world*" can be used referentially (i.e. when it is used to identify a particular person to the hearer), but can also be used attributively to say that whoever fits the description could lift the rock. It is rare for proper nouns and personal pronouns to be used attributively. However, contrary to what Fawcett says [Fawcett 1986, p.86], it may happen, as in (1c) and (1d), both of which *may* be attributive uses ('whoever is President', 'whoever laughs last'). There is also the question of whether "Macbeth" in (1e) is an attributive use, just as, presumably, "Duncan's murderer" is in (1f). In both cases the underlined definite NPs are being used essentially without

[2] In this case, it does not matter if it turns out that Duncan was not in fact murdered (he fell onto a chain-saw, accidentally or deliberately, which he had left running nearby). The absence from the model of someone to fit the description does not affect what Dogberry has said: he has predicated something about whatever kind of person could have intentionally done something of this kind to Duncan.

an object in mind: it is merely that the description is being attributed to the man being referred to by the subject NPs. Certainly Stalnaker [1972] describes (1e) and (1f) as attributive uses.

(1) a. "*Duncan's murderer is insane.*"
b. "*The strongest man in the world could lift that rock easily.*"
c. "*The President of the United States lives in the White House.*"
d. "*He who laughs last laughs longest.*"
e. "*That man/He is Macbeth.*"
f. "*That man/He is Duncan's murderer.*"

There are two very difficult problems to overcome if a system is to handle attributive uses of definite NPs: representation and recognition. From the point of view of representation, it is not at all clear what is needed. I believe that there are, in fact, a number of quite different examples collected under the attributive label. I give possible truth-conditions for (1a) to (1f) in (2a) to (2f) below (without worrying too much about any niceties of the logical forms here). (1a) and (1b) would be true in a model if the model's domain of discourse contains a denotation for the definite NPs, hence I have used a simple existential to capture this ((2a) and (2b)): an existential says some individual must have these properties but we do not know which one. (1c) seems to need some form of intensional representation: in (2c) I have quantified over times rather than possible worlds; the statement is true if whoever is President at time t lives in the White House at time t. (1d) seems like a simple universally quantified conditional (2d): if you are the one to laugh last, then you laugh longest. And (1e) and (1f) simply ascribe properties to an entity picked out by a referring expression ((2e) and (2f)):

(2) a. $\exists!x\ (\texttt{MURDERER-OF-DUNCAN}(x) \wedge \texttt{INSANE}(x))$
(where $\exists!x$ is 'there exists a unique x')

b. $\exists!x\ (\texttt{WORLDS-STRONGEST-MAN}(x) \wedge$
$\texttt{COULD-LIFT-EASILY}(x,\ \texttt{Rock1}))$

c. $\forall t \exists!x\ (\texttt{PRESIDENT-OF-US}(x,t) \supset$
$\texttt{LIVES-IN}(x,\ \texttt{WhiteHouse},\ t))$

d. $\forall x\ (\texttt{LAUGHS-LAST}(x) \supset \texttt{LAUGHS-LONGEST}(x))$

e. $\texttt{MACBETH(M)}$ (where $\texttt{M}$ is the referent of "That man/He")

f. $\texttt{MURDERER-OF-DUNCAN(M)}$ (likewise)

There is also evidence that although attributive uses do not pick out individuals, they might, in some cases, need to introduce entities into the discourse model. This is because they can act as anaphoric antecedents:

(3) "*The strongest man in the world could lift that rock. He could also carry an ox over his shoulder.*"

If such entities are to be put into the discourse model to allow "he" and "his" to pick up antecedents then they cannot be simple constants of the logic since they must receive a different model-theoretic interpretation (denotation) from other entities in the discourse model. (Again one is reminded of DRT).

The other problem with attributive uses is knowing when something is being used attributively and not referentially. There are no definitive syntactic clues to detect this, so contextual knowledge is important. A naïve solution would be to say that we should first treat a definite NP as being used referentially. Then, if an individual referent cannot be found, we should treat the NP as an attributive use. This might work occasionally but in general is inadequate. First, there are a number of other reasons why a referential construal might fail (see Section 7.5): it does not follow that we can always treat failure as indicative of an attributive use. Secondly, there are cases where a referential use might indeed succeed but where contextual clues suggest the speaker intended an attributive use. For example, (4) has a referential use (George Bush lives in the White House) and an attributive use (whoever holds the title gets to live in the White House):

(4) "*The President of the United States lives in the White House.*"

The contextual clues are more subtle than simple failure of a referential reading; for example, the definite NP in (4) *might* be more likely to be attributive in an encyclopaedia entry about U.S. politics.

These two problems are so difficult that this work does not attempt to make any real contribution to the issue of attributive uses of definite NPs.

- **Generic Uses**

In generic uses, the definite NP does not pick out an individual, nor does it strictly predicate something about whichever particular individual fits the description. Rather it gives us a property of all objects that presently are, have been and will be of that type. An example is: "*The camel is the ship of the desert.*" which tells us something about camels past, present and future.

T42 does not cater for generic uses as they have the same problems as attributive uses. First there are difficulties with representation: generic uses do not pick out individuals but they can be used as anaphoric antecedents ("*The camel is the ship of the desert: it sails across the sand, although going by aeroplane is more comfortable.*"). Secondly, they are not recognisable by their form.

The rest of this chapter explores the referential use in more detail, and the rest of the thesis relates presuppositions to this referential use.

5.3 Referential Uses of Definite NPs

In Chapter 4 I looked at processing (5a), "the portrait in the casket", in a context in which P1 was the discourse model entity representing the portrait in the casket. I said that T42 could at the close of processing this NP produce either the logical form (5b) or it could deliver the referent (5c):

(5) a. "The portrait in the casket ..."
b. $\exists x_0 \exists x_4$ (PORTRAIT(x_0) $\wedge$ IN(x_0, x_4) $\wedge$ CASKET(x_4))[3].
c. P1, i.e. PORTRAIT(P1) $\wedge$ IN(P1, C1) $\wedge$ CASKET(C1)

(5b) is obtained by simply accumulating constraints and not bothering to consult the discourse model. (5c) is obtained by consulting the discourse model and using the entities retrieved to solve a Constraint Satisfaction Problem (CSP), ultimately yielding a referent for the NP (if the CSP is solved).

This suggests that T42 may be run in one of two modes, which for the purposes of this section I will label A and B:

- Mode A: With the constraint satisfier not consulting the discourse model, the constraint satisfier merely accumulates constraints to produce a fragment of logical form.

- Mode B: With the constraint satisfier consulting the discourse model and retrieving entities from it, it should end up with a discourse model entity representing the referent of the NP.

Suppose T42 is used to process an utterance of a whole *sentence*, rather than just an NP, again different logical forms can result according to the mode used. For example, for the sentence (6a), mode A processing (discourse model not consulted) would give logical form (6b), but mode B processing (consulting the discourse model, and assuming the discourse model contains PRINCE-OF-DENMARK(Hamlet)) would give logical form (6c). (I ignore here the details of how the verb phrase is actually processed — see Chapter 6).

(6) a. "The Prince of Denmark smiled."
b. $\exists x$ (PRINCE-OF-DENMARK(x) $\wedge$ SMILED(x))
c. PRINCE-OF-DENMARK(Hamlet) $\wedge$ SMILED(Hamlet)

Both (6a) and (6b) entail (7):

(7) $\exists x$ PRINCE-OF-DENMARK(x)

((6b) using $\wedge$-elimination and (6c) using existential generalisation). If (7) is false, i.e. $\neg\exists x$ PRINCE-OF-DENMARK(x) is true, then (6b) would be false since (7) is an entailment of (6b).

But for the results of mode B processing, i.e. (6c), where the constraint satisfier was able to consult the discourse model, things are somewhat different. In this mode, processing a definite NP such as "the Prince of Denmark", T42 tries to solve a CSP using discourse model entities. Thus there is a *demand* that a suitable discourse model entity be found as the referent of this phrase. If such an entity cannot be found, the CSP will be unsatisfiable. And so only if such a referent can be found will T42 be able to produce (6c). In other words, it is a *precondition* on producing reading (6c) that PRINCE-OF-DENMARK(Hamlet) is in the *discourse model*. If

[3] Ignoring the uniqueness constraint on the definite NP.

PRINCE-OF-DENMARK(Hamlet) is not in the discourse model, (6c) could not be produced: this reading of the utterance would not 'come off'. I used two phrases in this description: a "demand" for a discourse model entity and a "precondition" that a discourse model entity be found. We will see in Chapter 7 when I look at other sorts of presuppositions that this is what I mean by a presupposition.

I can relate these ideas to Russell and Strawson's work and their debate on definite NPs. For Russell [1905 (1975)], (8a) below *entails* (8c) and, under one of its readings[4], (8b) *entails* (8c). For Strawson, (8a) and (8b) *presuppose* (8c). Strawson said "Statement A presupposes statement B iff B is a precondition of the truth or falsity of A." [Strawson 1952, p.187]. We will consider later whether Strawson's notion of "precondition" in this definition is the one I have just introduced. If (8c) is false, i.e. there is no Prince of Denmark, then for Russell (8a) and, under one of its readings, (8b) will be false. But for Strawson, falsity of (8c) results in (8a) and (8b) lacking a truth-value.

(8) a. The Prince of Denmark smiled.[5]
    b. The Prince of Denmark didn't smile.
    c. 'There is a Prince of Denmark.'

The heart of the disagreement between Russell and Strawson lies in understanding what type of objects they are investigating: is it propositions, sentences, utterances, statements or assertions? Strawson says that Russell is concerned with "sentences" whereas he (Strawson) is concerned with "statements". Now, Strawson is simplifying here: sentences are not what Russell deals with since sentences cannot be said to be true or false, nor do they have entailments. Russell makes it clear that the objects he deals with *will* be either true or false and *will* have entailments. Therefore, Russell must be concerned with *propositions*, which have both of these properties.

Strawson claims to be dealing with "statements". Nowhere does he make clear what he means by this term. All we know is that statements may be true or false or may lack a truth-value. There are two interpretations of Strawson on this point. One, discussed in [Garner 1971], says that Strawson's statements are illocutionary objects. In this sense, statements stand in contrast to other illocutionary objects such as questions, commands, promises, etc. If this view is correct, another word for "statement" here might be "assertion". One (in my opinion, unfortunate) consequence of this view is that questions, promises and illocutionary objects other than assertions are defined as not having presuppositions. In one place, my reading of Strawson supports Garner's view. Strawson says that the objects one uses in telling stories are *not* statements; that is, sentences in stories are *not being used* as statements. In stories, the felicity conditions for statements are not satisfied: the sentences are not statements and cannot be assessed for truth or falsity. They also therefore cannot have presuppositions [Strawson 1950].

[4] The narrow-scope negation reading: see Chapter 8 for a fuller explanation.

[5] I have not used my conventions to indicate whether (8a) and (8b) are propositions, sentences or utterances since this is the question under discussion in the text.

Kempson [1975, pp.51-52], however, prefers an alternative interpretation. She believes Strawson's statements are "abstract objects". Strawson is not using the term "statement" to contrast with questions, promises and other illocutionary objects. Rather, Strawson's statements are *propositions.* Kempson concedes that, in places, Strawson does seem to conflate a view of statements as illocutionary objects and as abstract objects (propositions). I prefer Kempson's view that Strawson is really dealing with "abstract objects" (propositions) and believe that the evidence (apart from the aberration about statements in stories mentioned in the previous paragraph) is on her side. In this case, other illocutionary objects (questions, promises, etc.) can have presuppositions.

It is also not clear what Strawson means when he says that a statement will be neither true nor false when one of its presuppositions is false. There are three interpretations of this:

1. No proposition gets produced (the reading fails).

2. A proposition is produced but it lacks a truth-value.

3. A proposition is produced but it has a third truth-value representing that its truth is unknown.

Logicians have preferred options (2) and (3). I prefer case (1), i.e. that the reading does not 'come off', and believe that this is consistent with the following quote from Strawson: "... the question of whether it's true or false simply doesn't arise." [Strawson 1950, p.330].

These interpretation problems about Strawson cannot be settled. Therefore, merely in order to make my own and Strawson's positions compatible, I will proceed as if by "statements" Strawson means propositions and as if Strawson intends that presupposition failure should result in the failure to produce a reading (option (1)).

This means that I have concluded that Russell deals with propositions and Strawson deals with propositions. Obviously this gets me nowhere in accounting for their different theories. The resolution of this, I believe, lies in determining what it is that Russell's and Strawson's propositions represent. For Russell, a proposition is a *sentence* reading, i.e. the truth-conditional content of a sentence, independent of context. For Strawson, a proposition is an *utterance* reading, i.e. the meaning of the uttered sentence in a particular context with discourse referents filled in with reference to the context of utterance. With these construals of Russell's and Strawson's positions, I have related Russell's view to mode A processing in T42 (where the discourse model is not consulted) and have related Strawson's view to mode B processing in T42 (where the discourse model is consulted). Mode A, we have seen, produces (6b) for (6a), as I think Russell would (ignoring the uniqueness constraint). Mode B would produce (6c) for (6a) (assuming `PRINCE-OF-DENMARK(Hamlet)` is in the discourse model), and this is compatible with Strawson's view (except that Strawson would probably dispute that natural language can be given logical analyses at all). If $\neg\exists x$ `PRINCE-OF-DENMARK(`x`)` then (6b) will be false in accord with Russell. If

PRINCE-OF-DENMARK(Hamlet) is not in the discourse model, then (6c) would not 'come off'.

Mode A processing and Russell's existential analysis give "sentence processing": they look at sentences independent of context and give a reading for these. Donnellan [1966] says that it is possible that Russell's analysis corresponds to attributive uses of definite NPs. Certainly, of the attributive uses I identified in the previous section, two of them, (1a) and (1b), did have Russell style truth-conditions.

I have related mode B processing in T42 to Strawson's ideas on definite NPs and to the referential use of definite NPs. We are both dealing with utterances (sentences in context) and both use the notion of a special kind of inference, a precondition, i.e. a demand which must be satisfied for a reading to come off, and these demands we call presuppositions. However, there is a difference between us.

I indicated in Chapter 3 that I follow Stalnaker in drawing a distinction between the context (discourse model) and possible worlds. Presuppositions must be satisfied by the discourse model. The discourse model contains a set of propositions and so determines a set of possible worlds. These possible worlds need not include the 'real' world (or whichever world against which the discourse is conventionally assumed to be tested for truth). Hence a presupposition can be satisfied by the discourse model but be false in the real world, or can fail to be satisfied by the discourse model but be true in the real world. Strawson, on the other hand, has no separate notion of context (or, if he does I presume that he takes it to be consistent with the real world). Hence for him testing a presupposition means testing it for truth in the real world (or whichever world we might be using).

There are several advantages to separating context from possible worlds. It means that we can felicitously hold conversations about 'non-existent' objects, be they fictional (Hamlet), mythological (Pegasus) and perhaps even impossible (perpetual motion machines and round squares). All that this means is that one can felicitously hold a conversation about the Prince of Denmark, Pegasus, etc. since discourse model entities will be found for these definite NPs. T42 can use these entities to get readings such as PRINCE-OF-DENMARK(Hamlet) $\wedge$ SMILED(Hamlet).[6] If a reading using these objects is then assessed for truth against the real world, the reading will be false or undefined[7] since the object does not exist in that world. But in possible worlds where the objects do exist, propositions may be true or false as the case might be.[8]

Following on from this, my approach also gives a reasonable account of utterances of sentences such as "*The bogeyman exists.*". For Strawson (and for other theorists such as Gazdar [1979][9]), this utterance presupposes the

[6] There then probably needs to be a way of showing that the entities are fictional or legendary and so have no real world counterparts.

[7] I feel I can remain agnostic on this point.

[8] And, if we allow partial, underspecified worlds, as in [Fodor 1979], we may yet have propositions being undefined in certain worlds.

[9] See Gazdar's potential presupposition trigger, f_2. [Gazdar 1979, p.126]

same proposition as it asserts, namely that the bogeyman exists. If presuppositions are to be preconditions and to have some sort of 'background status', this seems anomalous. On my account, such an utterance presupposes only that there should be an entity, B say, in the discourse model that can be identified using the definite NP "the bogeyman". It then makes a metalinguistic statement about B. Hence there is no anomaly. I will return to these metalinguistic uses and explain them more fully in Chapter 8 when I look at the presuppositions of utterances of negative sentences.

T42's approach might lead to another advantage over Strawson: Donnellan points out that in referential uses of definite NPs, the description used to pick out the entity need not be accurate. He says that Strawson seems to have overlooked this. We cannot modify Strawson to take inaccurate reference into account. For example, if we are looking for the King of France to satisfy a presupposition, there is currently no such object in the world so the presupposition is false and the reading does not come off; we cannot pick up President Mitterand since it is not true that he is actually the King of France. T42 does not, in its present implementation, cater for inaccurate reference, but the distinction between possible worlds and the discourse model, and the idea that presuppositions should be satisfied in the discourse model, not in the real world, might provide a handle on the problem of inaccurate reference: it will not matter that there is no King of France in the real world provided Mitterand's discourse model entity is labelled in such a way as to show that the speaker believes that this would be a suitable descriptor for Mitterand, as reference can then be successful.

A final issue requiring comment is the role of "topic" and "focus" in presupposition. I include this because Strawson in later writing [Strawson 1963] moved slightly towards Russell's viewpoint: he said that, while it was generally the case that a statement would lack a truth-value if one of its presuppositions is false, he could see that in certain cases a false presupposition could, in line with Russell, lead to the statement being false. His revised position was: if the presupposition of a definite NP as the "topic" of an utterance should fail then, as he had always maintained, the statement would lack a truth-value. However, if the presupposition of a definite NP not occurring as the "topic" of an utterance should fail then the statement would be false rather than truth-valueless. Thus, if there is no Prince of Denmark, (9a) lacks a truth value but (9b) is false:

(9) a. "*The Prince of Denmark chose the play.*"
b. "*The play was chosen by the Prince of Denmark.*"

Here I am assuming that, as is often the case in English, the topic and subject coincide. Strawson offers no definition of topic. He goes only so far as to say that topic is a centre of interest in the statement and what the statement is about. Thus, he claims, we assess truth in a topic-centred way: failure of topic presuppositions will be more serious than failure of other presuppositions.

Of course the lack of topic definition leaves Strawson's view open to all sorts of criticism. Kempson [1975, p.88] shows its problems. It means, for example, that a passive sentence may be false while its active counterpart could lack a truth-value, and it means that a sentence may go from false to truth-valueless in different utterances if all that changes is the speaker's choice of topic. I will not be following Strawson in this direction.

::::

This chapter has continued to look at definite NPs. It has put the processing T42 does into context by relating it to referential and attributive uses of definite NPs and also by relating it to the work of Russell and Strawson. With T42 consulting its discourse model, there is a lot of commonality with Strawson's ideas: the demand for discourse model entities to satisfy an NP can be likened to Strawson's notion of presupposition as a precondition. However, T42 can satisfy presuppositions by entities in its discourse model: the presuppositions do not have to be true in the real world. This has the potential to give an account of non-existent objects such as those that appear in stories and of cases of inaccurate reference. It will also prove important for handling presupposition "cancellation" in negative utterances (Chapter 8).

The next chapter simply takes T42 and extends it so that it can process other parts of speech. This is a necessary precursor to looking at other presuppositions in Chapter 7.

Chapter 6

Extending T42: Given and New

Chapter 5 showed that T42's definite NP processing can be regarded as presuppositional. In order to be able to look at other types of presuppositions (e.g. those arising from factive verbs), T42 has to be able to process other parts of speech. In this chapter I show how I have extended T42 so that it can handle verb phrases and indefinite noun phrases. With these extensions T42 becomes capable of processing utterances of simple sentences and short texts. These extensions fit naturally into T42. You will recall from Chapter 3 that T42 comprises a categorial grammar parser which sends constraints to a constraint satisfier. The constraint satisfier treats the constraints as a Constraint Satisfaction Problem (CSP) where the values used to solve the CSP come from a discourse model. These two modules are interleaved, with the constraint satisfier telling the parser whether to continue or discontinue the current analysis. In moving to a system which can handle other parts of speech, all that changes are some of the details of the operation of the constraint satisfier: the values needed to solve the CSP now need not always be in the discourse model already, some of them may be newly-created. Which of these is the case will depend (roughly speaking) on whether the constraint satisfier is processing a definite NP or some other part of speech.

In fact, the constraint satisfier behaves differently on different constraints not according to whether they come from definite NPs or from other parts of speech, but according to whether they arise while processing "given" or "new" environments.[1] Much of this chapter can be seen as an attempt to characterise what is meant by "given" and "new". The importance of "given" environments will become clearer in the next chapter where items that begin "given" environments are equated with presupposition triggers.

In this chapter I start with a rough characterisation of "given" and "new" and explore successive refinements to it. Section 6.1 gives an initial definition in which definite NPs set up "given" environments: in processing these, the constraints are all used to retrieve entities from the discourse model; everything else is "new" and constraints are simply accumulated. This cor-

[1] "Given" and "new" in this thesis would seem to correspond to Heim's notions of "familiarity" and "novelty" respectively [Heim 1982].

responds to the approach taken in [Mellish 1985]. In Section 6.2 I draw the same boundary between "given" and "new" environments, but I change "new" environment processing so that new entities are *introduced* into the discourse model. This is a simple but necessary step for later ideas. Section 6.3 then refines the definition of "new" to allow consultation of the discourse model in "new" environments. Section 6.4 refines the definition of "given" by allowing additional properties to be ascribed to discourse model entities during the processing of "given" environments. Section 6.5 gives the final definitions of "given" and "new".

What I have just said about the contents of Sections 6.4 and 6.5 might have given the impression that the "given"/"new" distinction is somewhat eroded in T42. Section 6.5 shows that this is not so and gives a brief comparison with the TACITUS system [Hobbs 1986] which has virtually abandoned "given" and "new" distinctions. Finally, in Section 6.6 I make a brief comparison between the structures built by T42 and Kamp's Discourse Representation Theory [Kamp 1984]. There is an interesting distinction that makes T42's structure the more convenient one for NLP.

6.1 A Simple Approach to "Given" and "New" Environments

As was said in the introduction to this chapter, to examine verb phrases and parts of speech other than definite noun phrases, an understanding of "given" and "new" is needed. I shall not go into all the possible ways (most of them informal) of characterising the "given"/"new" distinction (see, e.g., [Chafe 1976] [Prince 1981] [Brown & Yule 1983]). These treatments seem to agree in essence but each has a difference of emphasis[2]. Roughly speaking, from the speaker's point of view "given" components are those bits of an utterance which the speaker takes to be mutual between her and her interlocutor; "new" components are not necessarily taken as shared in this fashion. I am not going to review these informal notions of "given" and "new". From the point of view of this thesis, "given" and "new" are technical notions defined in terms of the behaviour of T42. They may not be directly comparable to more pretheoretical notions of "given" and "new", not least because they are defined from the hearer's (T42's) point of view while many informal definitions take the speaker's point of view, but I believe that there is a plausible relationship between my technical definitions and the more informal notions.

Following Mellish, I have used the notion of analysing pieces of text within 'environments': "...the idea is that the current environment specifies whether a fragment of text under consideration is providing 'given' or 'new' information. An environment will normally extend over a complete noun phrase or clause." [Mellish 1985].

[2] Since most "definitions" are informal it is hard to know whether these differences are merely terminological or are more fundamental.

Initial Characterisation of "Given" and "New"

- A "given" environment is one for which a discourse model entity must be found in the discourse model. The constraints are used to find the entity whose properties in the model match the constraints.

- A "new" environment does not demand discourse model entities.

In other words, within a "given" environment, the hearer has a *demand* that a referent be found. The constraint satisfier must use constraints to retrieve discourse model entities during "given" environments, but during "new" environments it simply accumulates constraints. It is the parser's responsibility to indicate to the constraint satisfier which kind of environment it is currently in.

A speaker signals that she is taking some part of an utterance to be "given" by syntactic means, e.g. the use of a definite NP. From the hearer's point of view, on seeing a definite NP, he knows that the speaker is taking its referent to be "given" and so he must use the description to try to identify some entity. For other parts of speech, e.g. indefinite NPs, there is no such requirement that a referent be found. At this stage the following simplification is used (in line with [Mellish 1985]):

- Definite noun phrases (as defined in Chapter 5: proper nouns, NPs with the definite article, etc.) when used referentially refer to objects the speaker believes she and her interlocutor are both familiar with and have both given some salience to. In other words, definite NPs refer to discourse model entities and so they signal "given" environments.

- All other parts of speech such as indefinite NPs, verbs and those prepositional phrases which are not attached to definite NPs implicitly signal "new" environments.

The simple definitions given here are successively refined in the rest of this chapter.[3]

To implement this in T42, I have a flag called GIVEN. On the instruction of the parser the constraint satisfier (CS) sets the flag when a "given" environment is encountered. The definite article, proper names and pronouns set the flag. While the flag is set, the CS knows it must solve a CSP using discourse model entities in the fashion I showed in Chapter 4. The flag is unset on the instruction of the parser when the "given" environment ends. This happens when a reduction produces NP on top of the stack (i.e. syntactically closes

[3] I change the idea that indefinite NPs signal "new" environments but I keep in place the assumption that the definite article introduces "given" environments. Mellish [1985] says that functionals such as "the mass of the sphere" are counterexamples since the mass will not be "given". This is not a problem for T42: provided the sphere has already been introduced into the discourse model, its mass will be inferable (see Section 7.5.1). However a case I cannot handle, and which Mellish does not mention, is "the mass of a sphere", which begins with the definite article but goes on to introduce the sphere and hence its mass as new entities into the discourse model.

the noun phrase), provided cardinality constraints (UNIQUEness) are satisfied. This switches the constraint satisfier off. When the constraint satisfier is switched off, constraints are not used to query the discourse model; they are simply accumulated.

I will give an overview of processing an example using this approach. Consider processing "*The dog saw a man*" where the discourse model contains DOG(Crab), i.e. T42 is taking it to be mutual that the entity Crab can be referred to as a dog. The lexical entries needed for this example are:

word	*category*	*meaning*
"a"	$\mathrm{NP}x_4/\mathrm{N}x_4$	—
"man"	$\mathrm{N}x_5$	$\texttt{MAN}(x_5)$
"dog"	$\mathrm{N}x_1$	$\texttt{DOG}(x_1)$
"saw"	$(\mathrm{S}\backslash\mathrm{NP}x_3)/\mathrm{NP}x_2$	$\texttt{SAW}(x_2, x_3)$[4]
"the"	$\mathrm{NP}x_0/\mathrm{N}x_0$	$\star\texttt{UNIQUE}(x_0)$ $\star\texttt{GIVEN}(x_0)$

Note that I have included an extra pragmatic constraint, $\star\texttt{GIVEN}(x_0)$, in the meaning of "the".

Schematically, processing is as follows. When the parser processes "the", it will send the constraints for "the" to the CS. The new constraint I have included in the meaning of "the" ($\star\texttt{GIVEN}(x_0)$) tells the CS to set the GIVEN flag. Thus when "dog" is processed, the GIVEN flag will be "on" and so the CS will use DOG to query the discourse model. This query returns {Crab}. At this point the parser tells the CS that it has recognised a NP. This tells the CS to wake up the uniqueness test. The test is satisfied (there is only the one referent, Crab): the NP has been processed successfully. The "given" environment thus comes to an end so the CS switches off the GIVEN flag. This will mean that the rest of the utterance is processed without consulting the discourse model: constraints are merely accumulated and not used as queries.

In a simple run of T42 this would produce the following constraints:

$$\begin{array}{lllll} \star\texttt{UNIQUE}(x_0) & \star\texttt{GIVEN}(x_0) & \texttt{DOG}(x_1) & \texttt{EQUAL}(x_1, x_0) & \\ \texttt{SAW}(x_2, x_3) & \texttt{EQUAL}(x_2, x_0) & \texttt{EQUAL}(x_4, x_3) & \texttt{MAN}(x_5) & \texttt{EQUAL}(x_5, x_4) \end{array}$$

These express the following logical form:

$$\exists x \exists y\ (\texttt{DOG}(x) \wedge \texttt{SAW}(x, y) \wedge \texttt{MAN}(y))$$

But, some of the constraints ($\texttt{DOG}(x_1)$ and $\texttt{EQUAL}(x_1, x_0)$) were evaluated under the GIVEN flag with the constraint satisfier "on" so we also know that: $x_2, x_1, x_0 = \{\texttt{Crab}\}$, i.e.

$$\exists y\ (\texttt{DOG(Crab)} \wedge \texttt{SAW(Crab, } y\texttt{)} \wedge \texttt{MAN}(y))$$

Further it should be clear that DOG(Crab) is a redundant piece of logical form: it was in the discourse model before we started. Once we had evaluated

[4] I ignore tense and also questions about the adequacy of such a simple representation of the meaning of "saw".

the constraint $\texttt{DOG}(x_1)$ in the $\texttt{GIVEN}$ environment and found out that $x_1 = \{\texttt{Crab}\}$, then $\texttt{DOG}(x_1)$ had done its job. All we needed it for was to identify the dog being talked about. With this accomplished, it could have been thrown away:

$$\exists y\ (\texttt{SAW(Crab,}\ y) \wedge \texttt{MAN}(y))$$

This view of logical form for incremental systems where bits of logical form can be thrown away once they have been treated as instructions to get things from a knowledge base or discourse model is also mentioned by Pulman [1987]. Section 6.4 shows that care must be taken with this: not every constraint that appears in a "given" environment can be thrown away in this manner.

This example has been simplified because, in fact, the semantics of verbs in T42 is more complicated than it shows. The main difference is that I use event variables [Davidson 1967].[5] My motives for adding event variables are:

1. They allow a simple treatment of prepositional and adverbial phrase attachment to VPs.
2. They allow the explicit use of case roles for verbs much more easily. Instead of role being positional, it can be explicit.
3. The introduction of roles allows the easy incorporation of selection restrictions into the lexicon.
4. They allow me to handle VP anaphora (see next section) and help to make presupposition processing uniform with definite NP processing (see Chapter 7).

Thus the lexical entry for "saw" might now be:

word	*category*	*meaning*
"saw"	$(\mathrm{S}x_0\backslash\mathrm{NP}x_1)/\mathrm{NP}x_2$	$\texttt{SAW}(x_0)\ \texttt{AGT}(x_0, x_1)\ \texttt{PAT}(x_0, x_2)$ $\texttt{EVENT}(x_0)\ \texttt{ANIMATE}(x_1)\ \texttt{THING}(x_2)$

In this, x_0 is the event variable, $\texttt{AGT}$ and $\texttt{PAT}$ are roles, and $\texttt{EVENT}$, $\texttt{ANIMATE}$ and $\texttt{THING}$ are selection restrictions[6]. The full logical form for the example "The dog saw a man" will now therefore be

$$\exists y \exists z\ (\texttt{SAW}(z) \wedge \texttt{AGT}(z,\ \texttt{Crab}) \wedge \texttt{PAT}(z, y) \wedge \texttt{EVENT}(z)$$
$$\wedge\ \texttt{ANIMATE(Crab)} \wedge \texttt{THING}(y) \wedge \texttt{MAN}(y))$$

6.2 A Better Treatment of "New" Environments

In this section I continue to have definite NPs signalling "given" environments and everything else as "new". I also continue to have "given" environments setting up a demand for existing discourse model entities. However,

[5] In fact I follow [Pulman 1989] in introducing such variables for states as well as events.

[6] I will not show selection restrictions in this thesis any further.

in "new" environments, constraints are no longer simply accumulated: new discourse model entities will be created and the constraints will be taken as properties to be ascribed to these entities. The simple scheme for processing "new" environments described in the previous section (i.e. simply accumulating constraints) gives an essentially Russellian analysis for indefinite NPs, as it translates them as existentially quantified variables. There is a well-recognised problem with this analysis (see, e.g., [Chastain 1975], [Heim 1982]). This is: how is it that, if indefinite NPs translate to existentially quantified expressions, we are allowed to refer to them using pronouns and other definite NPs in subsequent text?

Consider the following examples:

(1) a. "*The dog saw a man and* $\left\{\begin{array}{l}\text{he}\\ \text{the man}\end{array}\right\}$ *saw the dog.*"

b. "*The dog saw a man.* $\left\{\begin{array}{l}\text{He}\\ \text{The man}\end{array}\right\}$ *saw the dog.*"

The first clause in both of these translates to

$$\exists x_0 \exists y_0\ (\texttt{SAW}(x_0) \wedge \texttt{AGT}(x_0,\ \texttt{Crab}) \wedge \texttt{PAT}(x_0, y_0) \wedge \texttt{MAN}(y_0))$$

But what about the second clause? If it is translated as

$$\exists x_1 \exists y_1\ (\texttt{SAW}(x_1) \wedge \texttt{AGT}(x_1, y_1) \wedge \texttt{PAT}(x_1,\ \texttt{Crab}) \wedge \texttt{MAN}(y_1))$$

we fail to capture the anaphoric relation: these two sentences of logic on their own would not adequately capture the meaning of (1a) and (1b) since they do not constrain the agent in (1b) to be equal to the patient in (1a). To capture this relation we could allow the scope of the existential to extend over both clauses:

$$\begin{array}{l}\exists x \exists y \exists z\ (\texttt{SAW}(x) \wedge \texttt{AGT}(x,\ \texttt{Crab}) \wedge \texttt{PAT}(x, z) \wedge \texttt{SAW}(y) \wedge \texttt{AGT}(y, z)\\ \wedge\ \texttt{PAT}(y,\ \texttt{Crab}) \wedge \texttt{MAN}(z))\end{array}$$

Taking this approach for (1b) has the effect of allowing quantification to extend over whole texts. This may be a tenable analysis but it does lead to problems that need to be overcome. One such problem ([Heim 1982, pp.9-14]) is that dialogues will need to be quantified over too and this may lead to contradictory truth-conditions. For example, consider the following exchange between A and B:

(2) A: "*The dog saw a man.*"
B: "*The dog didn't see a man; it saw a mouse.*"

This would be assigned the following (simplified) self-contradictory truth-conditions $\exists x \exists y (\texttt{MAN}(x) \wedge \texttt{THE-DOG-SAW}(x) \wedge \neg\ \texttt{THE-DOG-SAW}(x) \wedge \texttt{MOUSE}(y) \wedge \texttt{THE-DOG-SAW}(y))$. These may well be the truth-conditions of the dialogue as a whole but we really ought to be able to say something about the truth-conditions of its constituent individual utterances and we cannot.

An alternative solution is gaining currency. This alternative is used in T42. Its precursors are Kamp's Discourse Representation Theory (DRT) [Kamp 1984], Heim's file-change semantics [Heim 1982] and Webber's discourse models [Webber 1981]. We have already seen that T42 has a discourse model which holds contextual knowledge in the form of a set of entities and properties of those entities. We know that definite NPs demand retrieval of entities from the model. We have not yet said much about how entities get into the model. It seems obvious that indefinite NPs should introduce new entities into the discourse model. These will then be available for subsequent reference by definite NPs[7,8].

The same argument applies to the event variables for verbs. We are used to definite NPs referring to events introduced by earlier verb phrases. This holds no surprises since we know that verbs are easily nominalised:

(3) *"Bottom, the Weaver, was presented$_i$ with an acting award for his part in 'Pyramus and Thisbe' yesterday. The presentation$_i$ was made by Theseus, Duke of Athens. It$_i$ took place in the splendour of the royal palace, just after tea."*

For more complicated examples, see [Schuster 1988]. These too can be cases where instead of existentially quantifying over event variables, new entities are introduced into the discourse model.

So when the system is working in a "new" environment, new entities are *created.* The constraints received while processing this environment are treated as newly-ascribed properties of the entities. These new entities and properties are added to the discourse model. (More specifically, they are initially added to the immediate linguistic context (ILC), and are then moved to the non-immediate context (NIC) at the end of processing the utterance).

Creating new entities causes a complication in connection with n-ary constraints where $n > 1$, such as AGT and PAT. Suppose we are processing the following constraints: $\texttt{SAW}(x)$, $\texttt{AGT}(x, y)$ and $\texttt{PAT}(x, z)$ for an active sentence (so y will eventually be equal to the subject and z to the object of the sentence). First, we tackle $\texttt{SAW}(x)$. For this, we simply create a new entity, say EV0, as the satisfaction set of x and add SAW(EV0) to the ILC. But what about $\texttt{AGT}(x, y)$ and $\texttt{PAT}(x, z)$? What will the satisfaction sets of y and z be?

[7] I am ignoring in this thesis a number of other difficulties with indefinite NPs. In particular, Mellish [1985] points out that "...the number of objects referred to in an indefinite noun phrase may not be ascertainable on a purely local basis." [p. 37]. He gives the following examples:

> *"Small blocks, each of mass m, are clamped at the ends and at the centre of a light rod."*
> *"A wooden stool 2ft 2in which consists of ... a uniform vertical leg at each corner."*

The underlined indefinites probably refer to three blocks and three(?) legs respectively.

There are also generic and predicative uses of indefinite NPs (e.g. *"Crab is a dog."*) for which entities should not be created. All of these cases are beyond the scope of this dissertation.

[8] Entities are introduced by means other than language: things that both participants can see or touch or smell can enter the model in ways I leave undefined.

We will not know the satisfaction set of y until we do a reduction between the subject NP and the verb, and we obviously do not know what the satisfaction set of z should be because we have not shifted in the object NP yet.

The object NP is the more difficult of these cases. What should we put as the satisfaction set of z? One solution might be to make the satisfaction set of z contain all the entities known to the discourse model[9], i.e. say that the verb's patient, z, could be any of the entities we already know about. Obviously this is no good. The patient may well be a brand new entity introduced by an indefinite NP. Another possible solution would be to create a new entity and put this into the satisfaction set of z. Clearly this is going to cause trouble if the object NP turns out to be a definite NP, which should pick up an existing entity. This existing entity and the newly created one will not be equatable.

Fortunately, there is another option. This is to leave z's satisfaction set unresolved until after the object NP has been processed. Instead of looking entities up or creating new entities, the set is simply left as constraints that have yet to be evaluated. In the case of z, we would leave z's satisfaction set as $z = \{z' : \exists x\ (\texttt{SAW}(x) \wedge \texttt{PAT}(x, z'))\}$. The same approach can be taken with y too, i.e. $y = \{y' : \exists x\ (\texttt{SAW}(x) \wedge \texttt{AGT}(x, y'))\}$. This does not suffer from the problems of the other approaches. z and y can be ultimately resolved to any new or existing entity which satisfies their constraints. I have therefore adopted this solution.

If I take into account the developments made in this section, processing the example introduced in Section 6.1, i.e. "*The dog saw a man*", would now involve the following:

1. Shift in "the", $\mathrm{NP}x_0/\mathrm{N}x_0$. Store $\star\texttt{UNIQUE}(x_0)$ and on the instruction of $\star\texttt{GIVEN}(x_0)$ set the `GIVEN` flag.

2. Shift in "dog", $\mathrm{N}x_1$. Its semantics are $\texttt{DOG}(x_1)$. Since the `GIVEN` flag is set, retrieve a satisfaction set for x_1, i.e. $x_1 = \{\texttt{Crab}\}$.

3. Reduce $\mathrm{NP}x_0/\mathrm{N}x_0$ and $\mathrm{N}x_1$ to $\mathrm{NP}x_0$, generating $\texttt{EQUAL}(x_1, x_0)$, i.e. $x_0, x_1 = \{\texttt{Crab}\}$. We have recognised a noun phrase, and uniqueness is satisfied, so unset the flag.

4. Subject type-raise $\mathrm{NP}x_0$ to $\mathrm{S}x_2/(\mathrm{S}x_2\backslash\mathrm{NP}x_0)$.

5. Shift in "saw", $(\mathrm{S}x_3\backslash\mathrm{NP}x_4)/\mathrm{NP}x_5$ with semantics $\texttt{SAW}(x_3)$, $\texttt{AGT}(x_3, x_4)$ and $\texttt{PAT}(x_3, x_5)$. `GIVEN` is not set, but instead of simply accumulating constraints we now create new discourse model entities. Create a new entity for x_3, i.e. $x_3 = \{\texttt{EVO}\}$. Since x_4 and x_5 appear in n-ary constraints where $n > 1$, they are left as $x_4 = \{x_4' : \exists x\ (\texttt{SAW}(x) \wedge \texttt{AGT}(x, x_4'))\}$ and $x_5 = \{x_5' : \exists x\ (\texttt{SAW}(x) \wedge \texttt{PAT}(x, x_5'))\}$.

6. Reduce $\mathrm{S}x_2/(\mathrm{S}x_2\backslash\mathrm{NP}x_0)$ and $(\mathrm{S}x_3\backslash\mathrm{NP}x_4)/\mathrm{NP}x_5$ to $\mathrm{S}x_2/\mathrm{NP}x_5$, generating $\texttt{EQUAL}(x_3, x_2)$ and $\texttt{EQUAL}(x_4, x_0)$. This gives us $x_3, x_2 = \{\texttt{EVO}\}$ and it also resolves x_4 to $x_4, x_1, x_0 = \{\texttt{Crab}\}$.

7. Shift in "a", $\mathrm{NP}x_6/\mathrm{N}x_6$.

[9] Or, perhaps, all those which are salient enough in some sense.

8. Reduce Sx_2/NPx_5 and NPx_6/Nx_6 to Sx_2/Nx_6, generating $\mathtt{EQUAL}(x_6, x_5)$. This does not help us resolve x_5 any further, i.e. $x_5, x_6 = \{x'_5 : \exists x\ (\mathtt{SAW}(x) \wedge \mathtt{PAT}(x, x'_5))\}$.

9. Shift in "man", Nx_7, with semantics $\mathtt{MAN}(x_7)$. Create a new entity for x_7, i.e. $x_7 = \{\mathtt{M}\}$.

10. Reduce Sx_2/Nx_6 and Nx_7 to Sx_2, generating $\mathtt{EQUAL}(x_7, x_6)$. This does resolve x_5: $x_5, x_6, x_7 = \{\mathtt{M}\}$.

All the input has been processed and an S has been recognised. We can still read off an existential translation, namely:

$$\exists x \exists y\ (\mathtt{SAW}(x) \wedge \mathtt{AGT}(x,\ \mathtt{Crab}) \wedge \mathtt{PAT}(x, y) \wedge \mathtt{MAN}(y))$$

but we can also read off the following, where we use both the actual entities found for the definite NP and those created for the indefinite NP and the verb:

$$\mathtt{DOG(Crab)} \wedge \mathtt{SAW(EVO)} \wedge \mathtt{AGT(EVO,\ Crab)} \wedge \mathtt{PAT(EVO,\ M)} \wedge \mathtt{MAN(M)}$$

In this EVO and M are the new entities which, along with their properties, will be moved from the ILC into the NIC now the utterance has been completed.

If the text continues with "*He saw the dog.*" or "*The man saw the dog.*" as in example (1), the GIVEN flag will be set while processing "he" or "the man" and this demands a discourse model entity. This can now be found since we have created the entity M.

It is worth at this point indicating the *naïveté* of the view that indefinites and verbs *signal* "new" environments, i.e. can switch *off* the GIVEN flag. This is not straightforwardly so: if we had a situation where indefinite NPs or verbs could switch off the GIVEN flag, then examples of restrictive relative clauses attached to definite NPs would be handled incorrectly. For example, in (4):

(4) "*The horse (which was) raced past a barn died.*"

to allow a new racing event or a new barn to be created would be wrong. Rather, it is clear that the racing event and the barn should already be in the discourse model. Utterance of (4) does not introduce a new racing event or barn. This is why indefinite NPs and verbs cannot switch off "given" environments: only the successful syntactic closure of the "given" constituent switches off the "given" environment.

T42 would process example (4) as follows. A "given" environment is triggered by the word "the" and this sets the GIVEN flag. Hence the constraint satisfier asks for all horses to be retrieved from the discourse model. In the light of the argument in Chapter 4 (describing the Principle of Referential Failure), this utterance would only occur in a context in which there was more than one horse that could be talked about. The reason why the speaker has given a restrictive relative clause is to enable the hearer to pick out one of the horses, namely the one which was raced past a barn. So the NP "the horse" cannot be closed as its uniqueness constraint would be violated. It is left open to have some modifying phrase attached to it. In Chapter 4 the modifying phrase was a PP, but here it is a relative clause. Because the NP has not been closed, the `GIVEN` flag has not been unset, i.e. we are still in a "given" environment. Hence when "raced past" is processed, because the `GIVEN` flag is set, a new entity is not created; rather, any racing event entities are retrieved from the discourse model. These are restricted to racing events whose patients are horses. Then the discourse model is asked for any barns it knows about. Although this is indefinite, the `GIVEN` flag tells the constraint satisfier to retrieve, not create. The barns are restricted to those which had horses raced past them. At this point with a full relative clause processed, the parser will attempt to close the NP again. This time, we assume, the uniqueness constraint is satisfied (if not the utterance will be unacceptable as there are no more modifying phrases to further refine the description). Only now is the `GIVEN` flag unset. The rest of the utterance is processed in a "new" environment: an event entity will be created for the dying event with the horse that was raced past a barn as its argument.

This does raise the question: what difference is there between a definite and indefinite NP? One major difference is that definite NPs can trigger "given" environments but indefinite NPs never do. But given this there is still another difference that can be seen by considering (5a) and (5b):

(5) a. "*The horse (which was) raced past <u>the barn</u> died.*"
    b. "*The horse (which was) raced past <u>a barn</u> died.*"

In (5a) and (5b) both underlined NPs are processed in "given" environments and so both demand that discourse model entities be found. But they differ in precisely the other area where the lexical entries for definite and indefinite articles differ: the property of uniqueness. In (5a), not only must there be a unique horse which was raced past a barn, there must also be a unique barn that had a horse raced past it. In (5b) there must be a unique horse which was raced past a barn but this horse might in fact have been raced past several barns, i.e in (5a) there are two uniqueness tests because there are two definite NPs, but in (5b) there is only one uniqueness test because there is only one definite NP.

In the light of the discussion given in this section, I can now give a reformulation of "given" and "new" (in fact, only the "new" part has changed):

Reformulated Characterisation of "Given" and "New"

- A "given" environment is one for which a discourse model entity must be found in the discourse model. The constraints are used to find the entity whose properties in the model match the constraints.

- A "new" environment is one for which a discourse model entity will be created and put into the discourse model. Constraints are ascribed to the entity as new properties.

Definite NPs, it is assumed, still trigger "given" environments. Verbs and indefinite NPs may occur within "given" environments (as in restrictive relative clauses attached to definite NPs) and so do not trigger "new" environments. Rather, "new" environments simply comprise anything outside of "given" environments. With this change to T42, whereby entities are created in "new" environments, T42 begins to take on many of the features of Discourse Representation Theory. The similarities and differences are examined in Section 6.6.

6.3 Refining the Notion of "New"

In Section 6.2 I finally characterised "new" environments as ones in which new entities would be created and added to the discourse model so that they would be available for anaphoric reference. A "new" environment is any part of an utterance that is outside of a "given" environment. Often a "new" environment encompasses verbs and indefinite NPs, but not definite NPs. However, I did give cases where verbs and indefinite NPs occur within a definite NP (as part of a restrictive relative clause) and so were in "given" environments. In this section, I look at verbs and indefinite NPs that are not within definite NPs — they are in "new" environments, but nonetheless they seem to be coreferential or anaphoric (in its broad NLP sense) to something. I give some examples and modify T42 to take these cases into account.

My first examples are cases where *two verb phrases* can be taken to be coreferential. This necessitates a change to T42 since it suggests that verb phrases in "new" environments do not always introduce new entities: they can pick up existing referents anaphorically. This situation occurs particularly when two utterances are connected by the "coherence relation" of "elaboration" [Hobbs 1979][10], i.e. where one utterance is an expansion of another, e.g.:

(6) "*Cressida hit Troilus. She bashed him on the nose.*"

Here I would say that the hitting event in the first utterance is being referred to again in the second utterance and is having some extra descriptional properties ascribed to it (in this case, something about the 'manner' in which Cressida hit Troilus —a bashing— and 'location' —on the nose). Obviously for such coreference to succeed there must be no conflict between the case role fillers of the verbs in each utterance, i.e. in (6), for "bashed" to be anaphoric to "hit", the agent of "bashed", i.e. "she", must be anaphoric to the agent of "hit", i.e. "Cressida", and, similarly, the patient of "bashed", i.e. "him", must be anaphoric to the patient of "hit", i.e. "Troilus". Since information about 'manner' and 'location' are not mentioned in the first utterance, it is compatible to introduce such information in the second.

To be able to handle this in T42, we can no longer simply create new entities for verb events in "new" environments. We must now have the option

[10] Alternative terminology: "discourse relation" of "respecification" [Reichman 1978], "rhetorical predicate" of "amplification" [McKeown 1985].

of making use of existing referents in the model. I have extended T42 as follows: verb phrase constraints now both create a new entity and look for any existing entities for their event variables in the discourse model. This is not to say that they are *expected* to find any suitable existing entities — the `GIVEN` flag is *not* set. The `GIVEN` flag is only set when there is a *requirement* that an existing entity be retrieved. There is no such requirement for verbs. But irrespective of the outcome of the search a new entity is also created.

- If the search succeeds in finding an entity then the variable's satisfaction set will contain both the new entity and all compatible existing entities from the discourse model.
- If no entity is found, the variable's satisfaction set will contain only the new entity.

The question we now need to ask is: does the same thing happen with indefinite NPs? I believe it does. The examples I have found come from [Dahl n.d.].[11] Dahl comes up with a set of indefinite NPs which she calls "specific attributives". She says these indefinite NPs appear in specific contexts but do not create new entities[12]. Dahl's examples are:

(7) "*Dr. Caius told me that exercise helps. Since I heard it from a doctor, I'm inclined to believe it.*"

(8) "*Launce and Panthino both volunteered to walk the dog. Since at least one person is willing to walk the dog, we don't have a problem.*"

(9) A: "*I'm afraid I miscalculated Jack Rugby's insulin dosage.*"
B: "*What happened?*"
A: "*He died.*"
B: "*So, a patient has finally died due to your carelessness.*"

The change I mentioned above, whereby in "new" environments entities are both retrieved if possible and created will handle these cases too.

6.4 Refining the Notion of "Given"

My characterisation of "given" environments has up to now been that they demand that discourse model entities be found. These environments are triggered by definite NPs and extend over the whole NP (including verbs and indefinite NPs in restrictive relative clauses). In the next chapter I

[11] Another candidate set of examples are cases where one uses an indefinite to be circumspect, e.g. "A certain man we know is a very heavy smoker.".

[12] She also says the context must be "given". She does not say what she means by this. Her use of "given" seems not to tie up with mine: I use it to mean that there is a *demand* that a referent be found; I think that she uses it to cover any case where we get coreference. You can see, particularly in example (9), that there is no demand that the entity be "given".

equate presupposition triggers with things that trigger "given" environments and so we see that it is not just definite NPs that trigger these environments.

Implicit in my characterisation of "given" so far is that, in order for an entity to represent a candidate referent for an NP, there must be a direct match between the constraints generated in processing the NP and the discourse model entity's properties. In other words, the constraints must be a subset of the properties. In this section, I modify this assumption: I show cases where only a partial match is necessary. Some of the constraints must match an entity's properties if that entity is to be considered as representing a candidate referent. But other constraints may ascribe new properties, provided these new properties are consistent with the entity's existing properties.

Examples of additional properties being conveyed in "given" environments are common in 'newspaper-speak' of which (10) is an invented example, but they can arise in more normal texts too (11):

(10) "*Robin Goodfellow, 32, was convicted of assaulting a 6-foot plastic banana yesterday. The deranged Sheffield steelworker was said to have had a fixation with plastic fruit since he was a boy.*"

(11) "*Malvolio, went to see a doctor about his rheumatic pains. She gave him a prescription.*"

In (10) we identify *the deranged Sheffield steelworker* with *Robin Goodfellow* using salience constraints and the fact that both refer to people (these are the properties that enable the discourse model entity to be identified)[13], but additional properties are also conveyed, i.e. that the referent is deranged and works in Sheffield as a steelworker. These constitute information that we did not know before and must be added to the discourse model. In (11), the gender of the doctor was unknown (not assumed to be male) and the pronoun provides enough information to pick out a referent and also tells us the doctor's sex.[14]

T42 has been changed to handle this sort of example. With the `GIVEN` flag set, constraints are still used to try to find discourse model entities. But they will return not just entities that satisfy the constraints but also any entities that are *compatible* or *consistent* with the constraints, i.e. where

[13] There are other factors that aid this identification, e.g. that assaulting plastic bananas could only be the act of a deranged person.

[14] Related to these examples are cases of what I will loosely refer to as epithets. Suppose you and I are sparring partners of old, with you in the role of the Conservative supporter and me in the role of the wishy-washy liberal. If I say to you:

"*I see the bloody woman wants to privatise the monarchy now.*"

you will know that I am referring to Mrs. Thatcher, for whom you have a discourse model entity `T`. But your discourse model does not describe the referent, `T`, as a "bloody woman" (quite the reverse!). Yet you manage to identify `T` in this exchange because you also have properties that say that *I* don't rate `T` very highly. Somehow you use these properties to get a referent. I am not sure how this works (e.g. how do you know when it is *your* beliefs about an object to use and not your beliefs about *my* beliefs) and so I have not implemented anything along these lines. (Happily, this example is now out of date).

the constraints do not conflict with other properties ascribed to the entities. Compatibility is defined as not violating mutual exclusion constraints on the type hierarchy. Thus T42 will correctly handle (12a) and will correctly fail on (12b):

(12) a. "*Malvolio saw a doctor. She cured him.*"
b. ??"*Malvolio saw a letter. She pleased him.*"

6.5 A Summary of "Given" and "New"

Final Characterisation of "Given" and "New"

- A "given" environment is one in which an entity in the discourse model *must* be found: it is the entity that must be "given". The constraints must enable us to do this but may also ascribe additional properties to the entity.
- A "new" environment is one in which an entity can be found or can be newly created. If one is found, then some of the properties will have been used to pick out the entity; the rest will ascribe additional properties.

The changes I have made to the "given"/"new" definition have not eroded the distinction: "given" environments still *demand* an entity and supply enough already known information to enable an entity to be identified; "new" environments make no such demand.

I will briefly compare this T42 approach to that used in the TACITUS project since I believe the "given"/"new" distinction has been eroded in TACITUS [Hobbs 1986], [Hobbs & Martin 1987], [Hobbs et al. 1988]. The TACITUS team say that "given" and "new" information are hard to tell apart in their universe of discourse, that of understanding fault reports, due to the lack of determiners in these reports. Their approach is to try to *prove* everything in the logical form. If any part of the logical form cannot be proved from the knowledge base, then it is *assumed.* This process of assumption within a proof they refer to as "abduction". This means that all (non-contradictory) logical forms will be felicitous since if they cannot be proved by deduction on the knowledge base, they will be assumed. This would seem to erode all differences between different parts of speech.

To try to reinstate these differences some things in TACITUS are more readily assumable than others (indefinite NPs will be most assumable, definite NPs least). Costs are assigned to the different parts of speech. These are the costs of assuming them in a proof. In order of increasing cost they are: indefinite NPs, nonnominals, bare nominals, definite NPs, i.e. it is cheap to create (assume) the information conveyed by an indefinite NP and expensive to assume the information conveyed by a definite NP. There are also costs on axioms to reflect how expensive it is to use a rule to do abduction. A whole proof will have a cost which will depend on the amount of proof and the assumptions needed. The idea is to prefer proofs which minimise cost. The cheaper proofs provide better interpretations in the sense that they are more minimal, they involved fewer assumptions and so tie in well with the knowledge already possessed.

There would seem to be problems with this approach. For example, it is not clear how this is going to work for non-declarative sentences or non-literal uses: will the cheapest proof still be the best? Also there is the practical problem of how to assign reasonable costs. I am not in a position to consider whether there are insufficient syntactic clues in the TACITUS application texts to enable a proper "given" and "new" distinction to be drawn. If there are not then this is probably because most descriptions refer to already known entities (presumably the texts report faults about machinery to someone who should know about the machinery). However in texts where the distinction can be made, then it should, as it is in T42. There remains a very clear distinction between "given" and "new": "given" *demands* an entity, "new" does not. Having said this, I am forced to reexamine the notion of "given" in the next chapter and consider the possibility of creating entities even in "given" environments. However, if this is to be done it will be done as a *last resort*, when all other attempts to find an existing entity have failed.

6.6 T42 and Discourse Representation Theory

I wish here to consider T42 as it has been described so far in relation to Kamp's Discourse Representation Theory (see, e.g., [Kamp 1984], [Sells 1985]). This will, I hope, make some of the issues surrounding the variables produced in constraints and the nature of discourse model entities clearer.

I will use the following simple text:

(13) "*The dog$_i$ saw a man$_j$. He$_j$ saw it$_i$.*"

Kamp provides rules for building Discourse Representation Structures (DRSs) from a syntactic parse of a sentence. A DRS consists of a set of *reference markers*[15] at the top and a set of *conditions* on those markers. (One DRS may also contain other DRSs embedded within it: I will ignore this for now). For the first sentence in (13) DRT would construct the DRS:

(14)
$$\begin{array}{|l|}\hline x\ y \\ \\ \mathtt{dog}(x) \\ \mathtt{saw}(x,y) \\ \mathtt{man}(y) \\ \hline\end{array}$$

DRSs are an intermediate level of semantic representation. Truth is then defined relative to a model (where "model" here does not refer to a discourse model but to a model as it is used in model-theoretic semantics). Informally:

[15] Kamp calls them "discourse referents". I prefer to follow [Sells 1985] here as I have already been using the word "referents" more informally.

"A sentence S, or discourse D, with representation m is true in a model M if and only if M is compatible with m; and compatibility of M with m ...can be defined as the existence of a *proper embedding* of m into M which, roughly speaking, preserves all the properties and relations which m specifies of the elements of its domain." [Kamp 1984, p.2]. So the above DRS is true with respect to a model if and only if we can find some function, f, which can pick out two individuals from the model, one for x and one for y, where those two individuals satisfy the conditions in the DRS.

Sells comments: "The apparent existential force of the definite and indefinite articles is not represented as quantification in the DRS; definites and indefinites are treated as expressions with free variables in this theory, and x and y are essentially those free variables. Rather, the existential force comes from the truth definition, from the requirement that *there is a* function that does what the satisfaction conditions require. Again, it is crucial that *definite and indefinite NPs are not quantificational* (i.e., they do not have inherent quantificational force)." [Sells 1985, p.4]. This idea that NPs set up reference markers (free variables) which receive existential force from the way truth is defined for DRSs is how DRT has allowed for the problem highlighted in Section 6.2, i.e. how we can capture anaphoric relations between objects introduced using indefinite NPs and subsequent definite references to those objects. We can see this by continuing our example. Further utterances extend the DRS. The second sentence in (13) would augment the DRS as follows:

(15)
$$\boxed{\begin{array}{l} x\ y\ z\ w \\ \\ \mathtt{dog}(x) \\ \mathtt{saw}(x,y) \\ \mathtt{man}(y) \\ \\ z = y \\ \mathtt{saw}(z,w) \\ w = x \end{array}}$$

The referring expressions, "he" and "it", set up new markers (z and w) with equality constraints to previous markers ($z = y$ and $w = x$) to capture the anaphoric relationships. We can now define the truth of this DRS with respect to the model as a new function, f', which picks out individuals for x, y, z and w that satisfy the conditions, noting that in this case the individuals picked out for y and z are the same, and likewise for w and x.[16]

DRT is an attractive theory since the solution it provides to the problem of anaphoric reference is natural and simple. However, one part of the theory is left as pretheoretical. It is a part that needs further definition if DRT is to be usable in computational systems. The omission (with reference to the

[16] There is more to DRT than I have described. In particular, one DRS may contain a set of other DRSs with some partial ordering defined on them. This is useful for conditionals and universals and in particular allows a treatment of 'donkey sentences'. The definition of truth has to be extended to cater for this and a definition of "accessibility" based on the partial ordering can be given to inhibit certain illegal anaphoric relations. I will describe this further in Chapter 9.

example) is that Kamp does not indicate how one might determine that z should be equated to y and not to x and w to x and not to y (or to some other marker). He talks of choosing a "suitable" marker and so leaves his treatment incomplete. This may not seem important since there is a lot of NLP literature on choosing anaphoric antecedents for referring expressions (e.g. using syntactic constraints, selection restrictions and preferences, focussing and salience and general world knowledge; see [Carter 1987] for a review of all this). However, a DRS level of representation alone cannot easily represent competing *candidate* antecedents about which different properties might be known. T42 has the extra level of representation needed for this. To show this I will draw parallels between DRT and T42 which bring out the fact that T42 has an extra level of indirection that DRT does not have which allows T42 to do reference resolution more easily.

Both T42 and DRT have free variables or reference markers. For T42 these are the variables of the constraint satisfaction problem, sent to the constraint satisfier by the parser and stored in the ILC. Both T42 and DRT then have conditions or constraints on these markers. If DRT's markers can be mapped to objects in the model with all conditions preserved in the model then the DRS is true. However, in T42 a reference marker (variable) will have a satisfaction set of discourse model entities. An entity in a satisfaction set might have been newly introduced to the discourse model if the marker was created in a "new" environment. But if the marker was created in a "given" environment, the set of entities in its satisfaction set will have been retrieved from those already existing in the discourse model.

Since satisfaction sets may contain several entities that are compatible with the constraints seen so far, T42 has a representation of referential ambiguity. As further constraints are applied, the satisfaction set might be pruned, this being the process of reference resolution. This pruning may be done using both properties (constraints) in the ILC and properties in the NIC (and also knowledge from the knowledge base). For example, suppose marker x has a constraint on it that it is a person and has a satisfaction set of two discourse model entities that are persons {A, B} and the discourse model tells us that A is a man and B a woman. If a later constraint on x (e.g. a selection restriction coming from a verb) insists that x be female then using the discourse model A can be ruled out. I believe that for a computational system it is important that we have a representation of referential ambiguity: T42, while having many similarities with DRT, also has this level of representation.

::::

I have now extended T42 to a system which can handle verb phrases, indefinite NPs, simple sentence and texts with anaphoric references in them, where originally all it could handle was definite NPs. These extensions have affected only the constraint satisfier. The rest of the processing 'philosophy' remains the same: each word is processed as much as the system allows before the next word is read in, and if constraints are violated the parser can be told to choose an alternative analysis. The "given"/"new" distinction I have drawn is important not only for handling simple definite and indefinite NPs, but also for presupposition processing. I will show in the next chapter that presupposition triggers and things that signal "given" environments are the same. Presupposition satisfaction corresponds to satisfying a "given" environment's demand for discourse model entities. That "given" environments may ascribe additional properties to their referents will prove useful when I look at the "projection problem" in Chapter 9.

Chapter 7

Presuppositions in T42

So far in this thesis I have explored definite noun phrase processing in the incremental system T42. I was able to relate this to Strawson's work on presupposition. A strong similarity was demonstrated as we both use the notion of a precondition. However, unlike Strawson, but following Stalnaker, I have drawn a distinction between context (the discourse model) and possible worlds. As a precursor to showing that these ideas can be extended to other sources of presupposition, Chapter 6 described changes made to T42 to enable it to process parts of speech other than definite NPs.

In this chapter, I explore further my conception of presupposition. In section 7.1 I look at other characterisations of presupposition and indicate that they are distinct from my notion. Section 7.2 attempts an (informal) definition of presupposition based on the ideas given in earlier chapters of the thesis. With this 'definition', Section 7.3 considers other parts of speech to determine which of them might be presuppositional, and in particular illustrates how one additional presupposition trigger, the factive verb, is easily incorporated into T42. Section 7.4 compares the idea that presuppositions are a special kind of inference, preconditions, with the idea that seems more prevalent in NLP that presuppositions are some other kind of inference. Finally Section 7.5 looks again at the issue of what to do should a presupposition fail.

7.1 Different Types of Presupposition

"Presupposition" is a label that can be given to speakers' assumptions, to various types of precondition, and to 'ordinary' inferences. In this section, I briefly look at some of these. Each is logically or computationally different from presupposition as viewed in this thesis and I mention them now in order to exclude them from further consideration.

7.1.1 Speakers' Assumptions

Stalnaker is the best known proponent of the viewpoint that presuppositions are speakers' assumptions. When distinguishing his notion of presupposition from others, he uses the phrase "pragmatic presupposition". This is a little unfortunate: most other types of presupposition (e.g. as inferences drawn from utterance readings rather than sentence readings) could equally well use this phrase. A better phrase is "speaker presupposition":

> "According to the *pragmatic* conception, presupposition is a propositional attitude, not a semantic relation. People, rather than sentences or propositions are said to have, or make, presuppositions in this sense. More generally, any participant in a linguistic context (a person, a group, an institution, perhaps a machine) may be the subject of a presupposition. Any proposition may be the object, or content of one." [Stalnaker 1972, p.387].

> "A speaker presupposes that p at a given moment in a conversation just in case he is disposed to act, in his linguistic behaviour, as if he takes the truth of p for granted, and as if he assumes that his audience recognizes that he is doing so." [Stalnaker 1973, p.448].

Roughly speaking, this definition says that a *speaker* presupposes a proposition p just in case she believes, assumes or acts as if the proposition belongs to the shared background information. This means that it is not necessary for a speaker to have said or done anything to indicate presupposition of p. The definition does not say that the speaker must believe the presupposition. A speaker presupposes a proposition merely by leading people to believe the proposition: she need not have any commitment to it herself. While I agree with Stalnaker that presuppositions need not be believed by the speaker or the hearer, I am only dealing with those presuppositions which a speaker directly reveals (and a hearer detects and checks for) by her utterances.

Stalnaker does recognise that there are such things as sentence or utterance presuppositions too, i.e. presuppositions triggered by the use of certain parts of speech. He rightly points out that, while there is substantial overlap, there may be cases of speaker presupposition which are not cases of sentence or utterance presupposition and *vice versa*. Thus this notion of speaker presupposition is different from, but may coexist with, my notion of presupposition.

7.1.2 Utterance Preconditions

While I believe that presuppositions are preconditions, for me they are preconditions on the hearer successfully understanding an utterance reading. Other writers seem to have used the term "presupposition" to convey the notion of a precondition for a speaker to produce a particular illocutionary object or produce a culturally acceptable utterance.

- **Felicity Conditions on Illocutionary Objects**

Utterances, as well as conveying meaning, perform actions. I shall concentrate on what Austin called their *illocutionary acts* [Austin 1962] (also called *speech acts*). Examples of illocutionary acts are stating something, promising something, ordering that something be done, requesting some information and so on. The result of an illocutionary act is the production of an illocutionary object, e.g. a statement, a promise, an order, a request and so on. Declarative sentences often make statements, interrogative sentences often request information and imperative sentences often order that something be done.

Each act has a set of *felicity conditions* associated with it, these being conditions that must be satisfied for an act of that kind to be achieved. For example, for (1a) to count as the act of making a promise, the felicity condition, (1b), must hold:

(1) a. "*I promise to write to you.*"
b. 'Speaker sincerely intends to write to the addressee.'

If it does not hold, then the utterance of (1a) does not constitute making a promise; it constitutes some other act, e.g. lying or joking.

Presuppositions should not be equated with these felicity conditions. For example, one might say that one "presupposition" of the instruction "*Open the door!*" would be that the speaker believes that the door is not already open. More generally, one should only instruct someone to make p true if one believes that p is not already true. But this is simply a felicity condition on the act of ordering. Giving felicity conditions the additional label "presuppositions" seems unnecessary. The conditions on successful production of a particular illocutionary object (felicity conditions) are quite different to the conditions for a hearer to understand successfully an utterance reading (my notion of presuppositions). To use the terminology of speech act theory, for me presuppositions are preconditions on determining the propositional content of an act; they are not conditions on the act constituting one act rather than another.

• Cultural Conditions on Utterances

In one of his definitions, Keenan [1971] says that presuppositions could be those utterance preconditions that go towards making an utterance 'appropriate' or 'culturally acceptable' in some way. His main examples are the conditions of use associated with honorifics and familiar forms of address (e.g. use of "tu" in French).

Again, these conditions on 'cultural acceptability' are quite different to my notion of presupposition, which characterises conditions for successfully getting some utterance reading.

With these broader kinds of "presupposition" ruled out of further consideration, I will now return to looking at the particular utterance preconditions which I believe are presuppositional.

7.2 A Characterisation of Presupposition

Certain words and phrases are presupposition triggers (e.g. the definite article, factive verbs, etc.). The simplest way to "define" my notion of presuppositions would be to give an extensional definition: a list of presupposition triggers and the presuppositions they trigger. I give such a list, albeit an incomplete one, in the next section. A precedent for this type of definition is the work of Gazdar [1979, pp.125-127][1].

However, I would like to attempt a more intensional definition, even if it is only an informal one, as follows:

> Presuppositions arise only in connection with utterance processing. For a hearer to be able to determine a reading for an utterance, certain presuppositions must be satisfied. These presuppositions are preconditions which demand that the hearer be able to identify an entity in his discourse model as the referent of a phrase.

This remains an informal characterisation rather than a definition: it does not make clear what it means for something to be in the discourse model, for example, and it might embrace certain phenomena that I would not want to regard as presuppositions. I am not going to attempt to give a tighter definition, but I will try to bring out a few further points that characterise presupposition and perhaps help to distinguish it from other phenomena.

1. It is not speakers, utterances or sentences that presuppose things. It is *utterance readings* that have presuppositions, where an utterance reading is a proposition. In order to produce an utterance reading, its presuppositions must have been satisfied.

2. Therefore, presuppositions are *preconditions* on the production of an utterance reading. The distinctions between these preconditional inferences and other types of inference are drawn out in Section 7.4.

3. Presuppositions are thus defined from the hearer's point of view. This is to say that, in processing an utterance made by the speaker, the *hearer* looks to see what presuppositions are triggered by an utterance reading and attempts to satisfy these against his discourse model. I am not saying they are things the speaker believes true, or believes the hearer believes true (although these might certainly be the case). Rather we look at them only from the hearer's viewpoint: as things that must be satisfied if a reading is to be produced.

4. If a presupposed precondition is not satisfied, the reading is ruled out. This does not rule out the possibility of there being other readings whose presuppositions are satisfied. If all readings are ruled out, the utterance is

[1] Indeed, all the formal and computational approaches to presupposition rely on such a list as the core of their definition (e.g. [Karttunen & Peters 1979], [Gunji 1981] and [Van der Sandt 1988]). Even Mercer [1987], who attempts a more intensional definition in terms of inferences that can be drawn from default rules (see Chapter 10), has to say within his definition that it only applies to certain default rules (those responsible for presuppositions) and these are defined extensionally.

perhaps contextually unacceptable (but see Section 7.5). For a presupposition to be satisfied requires only that appropriate entities be in the discourse model. This is not the same as requiring them to be true in the 'real' world (or whatever world against which propositions are being tested for truth). It need not matter that a presupposition is false in the real world: it can still be satisfied by things in the discourse model. Generally it would be true to say that if a presupposition *is* satisfied by the discourse model, and the utterance reading is true in a world, then the presupposition will be true in that world. (However, when, in Chapter 9, we look at utterances of complex sentences, we will see that presuppositions may be satisfied by entities put into immediate linguistic context within the utterance of a conditional sentence, for example, where the fact that these entities are introduced in a conditional means that even if the utterance reading is true in a world, the presupposition need not be an entailment).

5. The lexical semantics of words may contain presupposition triggers. The presupposition is triggered in *all* circumstances where the word is used. Lexical items, such as the definite article, whose lexical semantics contains the constraint ⋆GIVEN, are effectively presupposition triggers. The ⋆GIVEN constraint tells T42 that the discourse model *must* supply entities for this part of speech. These presuppositions are triggered and must be satisfied no matter where they occur, e.g. even in negative sentences (Chapter 8) and conditionals (Chapter 9).

6. Presuppositions have a 'background feel' or seem to have 'uncontroversial status' because they are preconditions, quite distinct from the actual assertion being made by an utterance.

These would seem to be the major properties of presuppositions. We are now in a position to consider what presupposition triggers other than definite NPs there might be.

7.3 Presuppositions of Other Parts of Speech

Up to now, I have concentrated on the presuppositional properties of definite NPs, as these were the focus of Russell's and Strawson's debate. In this section I consider some of the other archetypical cases of presupposition. I shall not attempt a comprehensive review. I will select a few examples from [Karttunen n.d.] and [Levinson 1983, pp.181-185], some of which I will agree are presuppositional and others which I will argue need an alternative treatment, though I will not attempt to give these alternative treatments.

To make distinctions like this, i.e. about which things are and which things are not presuppositional, really requires a definition of presupposition. In the previous section, I gave only a rough characterisation rather than a definition and pointed out that often "definitions" are only extensional anyway, i.e. to some extent these are presuppositions because I list them under the heading "Presuppositions".

All I have done here is to pick a few examples from the literature and say whether I think they are presuppositional on my conception of presuppositions. I do this only intuitively by trying to decide whether they could sensibly have a ⋆GIVEN constraint in their lexical entries and so demand the retrieval of discourse model entities. This is obviously not a rigorous way of making such judgements, but making these judgements more carefully is not really the purpose of this work. The purpose is to offer a computational mechanism that can handle a reasonable set of the examples adduced in the literature. However, where my judgement needs help, I invoke the standard "negation test" for presuppositions. I ask whether the same demand for a discourse model entity is made by utterances both of a positive sentence and of its corresponding negative. This test is open to criticism, not least because it has been claimed that utterances of negative sentences need not always share the presuppositions of their corresponding positive sentences (see Chapter 8). The test is therefore only a guide and not a definitive test.[2]

7.3.1 Presuppositional Cases

1. Definite NPs

I have already shown that definite NPs demand referents. They contain the ⋆GIVEN constraint in their lexical semantics, which initiates a "given" environment in which discourse model entities must be retrieved. Therefore, (2a) and (2b) presuppose (2c):

(2) a. "*Launce's dog is happy.*"
    b. "*Launce's dog is not happy.*"
    c. 'Launce owns a dog.'

Although I show presuppositions as propositions here, it should be remembered that such propositions are to be treated as instructions to find discourse model entities satisfying these propositions, i.e. in this case there is the presupposition that the discourse model should contain a dog owned by Launce.

In Chapter 4 I looked at the uniqueness constraint on definite NPs and suggested that it was not presuppositional, but should instead be accounted for as a conventional implicature.

2. Factive verbs

Factives are verbs which take sentential complements and which, informally speaking, demand the familiarity of what these complements denote. Examples are verbs such as "regret" and "know", in contrast to "believe" and "wish". Factive verbs will specify that their complements should be processed as "given" environments. Therefore, (3a) and (3b) presuppose (3c) (i.e. they presuppose that a referent for the hitting exists in the discourse model):

[2] In fact, in Chapter 8 I will propose that negatives *do* always share the presuppositions of their corresponding positives, that is they do always demand presupposition satisfaction against the discourse model, and I will give an alternative analysis of cases of presupposition "cancellation".

(3) a. "*Romeo regrets he hit Juliet.*"
b. "*Romeo doesn't regret he hit Juliet.*"
c. 'Romeo hit Juliet.'

This is where event variables become useful again. (3a) and (3b) presuppose that the discourse model contains an entity that is a hitting of Juliet by Romeo. The lexical entry for "regret" can set the `GIVEN` flag which will force a referent for the event variable of the complement to be retrieved from the discourse model. I give a full example of this in the next subsection.

Horton [1987, pp.59-60] disagrees with the above judgement. She says that (3a) and (3b) presuppose (4):

(4) $\texttt{Bel}_{Romeo}\texttt{(Romeo hit Juliet)}$

where `Bel` is the belief operator. That is, the presupposition is that Romeo believes he hit Juliet. I do not agree. There is no demand that (4) should be true. It seems to me that (4) might well be an entailment of (3a). To my mind it does not necessarily survive in (3b). By contrast (3c) does seem to be a precondition on getting a reading for both (3a) and (3b). Horton bases her argument on the following example:

(5) ?"*Parolles regrets he made a fool of himself, but he didn't really.*"

I find this infelicitous but Horton does not. If you do find it felicitous then maybe "regret" is not a factive after all. That might be so, but is not the sort of argument I want to develop here. Most have taken it as a factive and as presuppositional, e.g. [Kempson 1975], [Gazdar 1979], [Gunji 1981], and [Mercer 1987]. "Be aware" is most definitely a factive and is even more clearly infelicitous in an example similar to (5):

(6) ??"*Parolles was aware he had made a fool of himself, but he hadn't really.*"

3. Implicative verbs

Traditionally, it has been suggested that implicatives have two presuppositions, both usually specific to the actual verb. Here is an example using "manage", where (7c) and (7d) are the alleged presuppositions of both (7a) and (7b):

(7) a. "*Caliban managed to find a scrap to eat.*"
b. "*Caliban didn't manage to find a scrap to eat.*"
c. 'Caliban tried to find a scrap to eat.'
d. 'Caliban found it difficult to find a scrap to eat."

I agree that (7c) is a presupposition: both the positive and the negative utterances seems to imply that a 'trying' event should be found. But I do not think that (7d) is a presupposition. Rather I would think that it is an implicature (although I am not sure whether it is conversational or conventional).[3]

4. Aspectuals

The existence of a previous event entity is presupposed by certain verbs which describe changes of state:

(8) a. "*Lear stopped beating his fool.*"
b. "*Lear didn't stop beating his fool.*"
c. 'Lear had been beating his fool.'

5. Iteratives

Iteratives too can presuppose the existence of previous event entities:

(9) a. "*Lear hit his fool again.*"
b. "*Lear didn't hit his fool again.*"[4]
c. 'Lear had hit his fool before'

i.e. there is a demand that another hitting event be in the discourse model.

There are probably many other cases which are presuppositional. T42 has been extended to handle factives as explained in the next subsection.

7.3.2 Factives

The factive example I gave in the previous subsection was "*Romeo regrets he hit Juliet.*". This is only felicitous if the discourse model contains an entity representing a hitting of Juliet by Romeo. The lexical entry for "regret" in T42 would therefore be:

word	*category*	*meaning*
"regret"	$(\mathrm{S}x\backslash\mathrm{NP}y)/\mathrm{S}z$[5]	REGRET(x) AGT(x,y) ⋆GIVEN(z) MOBJ(x,z)

The variable, z, on the sentential object is supposed to denote whatever the sentential complement denotes, and MOBJ means mental object [Boguraev & Spärck Jones 1987]. I follow [Hobbs 1985] in letting the sentential object denote a simple event or state[6]. The main point for my purposes is that

[3] Similar to these, although not actually implicative verbs, are verbs such as "pass" and "fail", where "*Parolles passed the test.*" and "*Parolles didn't pass the test.*" allegedly presuppose that Parolles took the test. If these *are* presuppositions (and I am not convinced that they are), then they could be handled in a similar way to "manage": their lexical entries will include a demand that their agent attempted the test described by the object NP.

[4] This is not so clear-cut: there is an ambiguity to consider. (9b) can mean that what Lear did again was not hit his fool or that what Lear didn't do again is hit his fool. Only the latter has presupposition (9c). Determining whether this presupposition is satisfied would enable T42 to choose the correct reading in context.

[5] This is a verb phrase ($\mathrm{S}x\backslash\mathrm{NP}y$) with a sentential object, Sz. I am ignoring other subcategorisations that would be needed to handle "*Romeo regrets hitting Juliet.*", "*Romeo regrets the fact that he hit Juliet.*" and "*Romeo regrets that he hit Juliet.*".

[6] This is an instance of what Hobbs calls "ontological promiscuity" [Hobbs 1985].

factives place some demand on their second arguments. This demand is that the event denoted by the sentential complement be "given". This is shown by setting the GIVEN flag during processing of the complement. It can be seen that using event variables in the semantics in this way has been very useful as it has enabled presupposition satisfaction for factive verb complements to be just like definite NP processing: a discourse model entity must be found.[7]

For utterance (3a) if the discourse model contains ROMEO(R), JULIET(J), HIT(EV1), AGT(EV1, R) and PAT(EV1, J), i.e. that Romeo has hit Juliet, this would give:

$$\texttt{REGRET(EV0)} \wedge \texttt{AGT(EV0, R)} \wedge \texttt{MOBJ(EV0, EV1)} \wedge \texttt{ROMEO(R)} \wedge$$
$$\texttt{HIT(EV1)} \wedge \texttt{AGT(EV1, R)} \wedge \texttt{PAT(EV1, J)} \wedge \texttt{JULIET(J)}$$

I will show how this logical form would be produced.

We begin with "Romeo", which picks up {R} as its referent. Next we process "regrets". This is a verb and, as the GIVEN flag is not set, we both look an entity up and create a new entity. However, the search for an existing entity fails (our discourse model has no regrets!) and so the satisfaction set for the event variable will contain only the new entity, EV0, say. The agent of this event can be resolved to the subject NP referent, R.

But also in the lexical semantics of "regrets" are the constraints $\star$GIVEN(z) and MOBJ(x,z). Thus the GIVEN flag will be set and will remain set while we look up EV0's MOBJ. The MOBJ can be any event. There are two candidates: from the non-immediate context (NIC) we retrieve EV1, and from the immediate linguistic context (ILC) we retrieve EV0 itself. Remember that since the GIVEN flag is set we do not create a new entity. There is an axiom which says that an event cannot be its own MOBJ and this rules EV0 out, leaving MOBJ's satisfaction set as {EV1}.

We now read in "he" which picks up {R}. Then "hit" is constrained to be equal to the MOBJ and hence must have as its satisfaction set {EV1}. Finally, "Juliet" picks up {J} and we are done.

The new information is REGRET(EV1), AGT(EV1, R), and MOBJ(EV1, EV0). We see that the hitting event was picked up as the referent of the complement because the complement was processed in a "given" environment.

Note that if no hitting events had been in context or the hitting event in context had not been compatible with the one described in the complement (which had a male agent and Juliet as patient) then the utterance reading would not have come off. We can use this to sort out certain structural ambiguities in a fashion similar to our handling of PP attachment problems. For example, (10a) is ambiguous as shown in (10b) and (10c):

(10) a. "Romeo regrets he hit Juliet and Juliet cried."
b. 'Romeo regrets [[he hit Juliet] and [Juliet cried]].'
c. '[Romeo regrets he hit Juliet] and [Juliet cried].'

[7] See Chapter 8 footnote 13 for a brief discussion of how this might work for cases where the complement is a negative clause.

If the context tells us only that Romeo hit Juliet, then (10c) will be found as a reading since Juliet's crying would only be acceptable if processed as not falling within the "given" environment.[8]

• Indefinite NPs: A Problem Solved

Certain writers, such as Gazdar [1979, p.152] and Mercer [1987, pp.180-181], have noted that indefinite NPs can have 'anaphoric behaviour':

(11) a. "*Bassanio regrets he made* a deal."
b. "*If Dogberry came to the party, then the hostess must have been really glad that there was* a policeman *present.*"

The indefinite NP "a deal" would seem to be felicitous only if we already know of, i.e. have in our discourse model some knowledge of, some deal that was made by Bassanio, and yet this is an indefinite NP, which would not normally demand such an entity. Similarly, the indefinite NP "a policeman" can either be coreferential with Dogberry or with some other policeman in the context of utterance. Gazdar's and Mercer's theories cannot account for this data.

I have already given an explanation of similar cases in Chapter 6. There I discussed verbs and indefinite NPs in restrictive relative clauses attached to definite NPs. Since they are evaluated in "given" environments, T42 retrieves entities for them. Similarly, in the examples in (11), since the indefinites occur in the complement of factives they are processed in "given" environments (i.e. while the `GIVEN` flag is set), and so are forced to pick up discourse model entities. The `GIVEN` flag is set by the factive verb; there is nothing in the lexical entry of the indefinite article that would ever make it trigger presuppositions itself.

I will now examine three cases which others have described as presuppositional but which I believe do not satisfy the characterisation of presupposition and so require a different treatment.

7.3.3 Non-Presuppositional Cases

1. Verbs of Judging

Some (e.g. [Fillmore 1971, p.288]) have claimed that (12a) and (12b) should presuppose (12c):

(12) a. "*Stefano accused Trinculo of running away from danger.*"
b. "*Stefano didn't accuse Trinculo of running away from danger.*"
c. 'It is bad (for Trinculo) to run away from danger.'

One thing that is definitely not a presupposition is 'Trinculo ran away from danger' (consider "*Stefano wrongly accused Trinculo of running away from*

[8] In fact, while in principle this can be disambiguated in such a fashion, T42 cannot manage this because the parser's control strategy only finds analysis (10c) and not (10b) —see Section 3.3.1.

danger."). Horton [1987, pp.65-66] agrees and suggests that the correct presupposition is $\texttt{Bel}_{Stefano}$(Trinculo ran away from danger). Horton's proposal is clearly wrong: *"The police accused the Guildford four of bombing."* does not presuppose $\texttt{Bel}_{Police}$(The Guildford four did a bombing)[9].

(12c), too, is not a presupposition. There is certainly no demand that it be true. It is probably a generalised conversational implicature. Therefore, verbs such as "accuse" and others such as "criticise" probably have none of the presuppositions that have been associated with them.

2. Selection restrictions

While selection restrictions are constraints of some kind, they do not require the retrieval of a discourse model entity, only the checking of a relationship against a constraint. We also saw in Chapter 2 (Section 2.2.2) that selection restrictions may be violated in negative utterances and still leave an utterance reading that comes off.

3. Non-Criterial properties

Some (e.g. [Mercer 1987, pp.76-78]) have suggested that certain 'non-essential' parts of the meaning of a lexeme might be presuppositional. For example, the word "bachelor" might presuppose that the subject is male and adult. These are background properties of sorts, in contrast to the fact that the subject is unmarried which is the new information conveyed by describing someone as a bachelor.

Again these do not demand that an entity be found in the discourse model and so they cannot be presuppositional in my sense of the term. This is not to deny that there might be default inferences that can be drawn when a word such as "bachelor" is used.

7.4 Presuppositions are Preconditions

The previous sections of this chapter have argued that presuppositions can be viewed as preconditions. This will prove advantageous when looking at the presuppositions of utterances of negative and complex sentences (see Chapters 8 and 9). But I can even now show that not only may presuppositions be characterised as preconditions, it is actually preferable that this be done.

First I should explain what I mean when I say that presuppositions are preconditional inferences in contrast to 'ordinary' inferences. This is not a clear formal distinction. Rather, what I mean is that, while from a formal or logical point of view, presuppositions are inferences, in my theory and hence in T42, these inferences have an additional computational or procedural aspect to them, namely that they are to be tested for satisfaction against the discourse model, and getting an utterance reading is conditional on their satisfaction. Hence, when I contrast preconditional inferences (presuppositions) with ordinary inferences, I am not claiming that there is any formal difference in the nature of the inferred propositions, rather I am claiming

[9] I thank Ann Copestake for this example.

that there is a difference in the *use* to which preconditional inferences are put.

Taking presuppositions as preconditions is contrary to most existing formal and particularly computational treatments of presuppositions. In existing accounts, some process takes in some sentence or utterance reading, applies some inference rules (i.e. fires some presupposition triggers) and spits out a set of ordinary inferences — the presuppositions. (In negative and complex utterances, the set of presuppositions is often seen as needing to be filtered or modified before being spat out). But this leads to the question of what to do with these inferences once they have been generated. There are two issues here: what their status should be and what use they should be put to.

If you have generated a presupposition and treat it as an ordinary inference by simply adding it to the discourse model or knowledge base, does it have the same status or strength as asserted or entailed information? On the one hand, it seems that it does have at least the same strength as an entailment or assertion. Indeed it probably has greater strength: presuppositions are 'uncontroversial', 'background' information. On the other hand, as we shall see in the next chapter, it is claimed that the presuppositions of negative utterances can be "cancelled". What is more, this "cancellation" can happen after a seemingly arbitrary amount of intervening text.[10] We need to reconcile the idea that presuppositions are cancellable with the idea that they are quickly "consolidated" to the point where a hearer takes them as being possibly more incontrovertible than asserted and entailed propositions.

With presuppositions treated as ordinary inferences I believe this question is largely unresolved. But treating presuppositions as preconditional inferences seems to accommodate these notions: presuppositions have a 'strong feel' to them precisely because they are preconditions of utterance readings and must be satisfied. But a speaker may later explicitly indicate that while a presupposition is satisfied in the discourse model, she does not actually believe it to be true in a particular world (see the next chapter). In principle, this can take place an arbitrary amount of time after the presupposition was first established, provided the speaker has suitable credibility and as long as the speaker has not 'confirmed' the truth of the presupposition in the interim.

The second question to address is: if presuppositions are ordinary inferences, to what use can these inferences be put? What do we do with such inferences once we have drawn them? The answer so far given by those treating presuppositions as ordinary inferences is not convincing.

Kaplan [1982], Gibson [1986] and Mercer & Rosenberg [1984] all generate presuppositions as ordinary inferences in coöperative natural language interfaces to databases. These inferences are tested against database constraints.

[10] An example from [Gunji 1981] is:

> "Jack doesn't regret being bald. I can show this quite easily. Look at the man in the corner. He is Jack, though it's a bit dark to see him clearly. But look at him carefully. Now you know what I mean. He isn't bald; he only regrets being grey-haired."

If they do not match the constraints, the user has some misconception about the database structure. Weischedel *et. al.* [1978] compute "presuppositions" as inferences of student responses in a foreign language tutor system.[11]

Both of these uses basically put the same interpretation on presuppositions as I have, i.e. as things to be tested to see if the logical form produced (or being produced) is consistent with what is already known. The difference is that I would do such tests *as part of producing an utterance reading*, while in the work reported above they are only produced and tested *after* some sort of reading has been produced.

Thus, for me presuppositions are not things we have hanging around which we can test if the system is to be coöperative; they are essential to getting a correct utterance reading. This means that, not only do presuppositions play a crucial role in checking whether utterance readings come off, they may also guide the process of utterance reading production. We have seen what this means for prepositional phrase attachment, where our partial solution to this problem was to decide which of the two competing readings was most likely on the basis of presupposition satisfaction. This involved looking for referents of NPs and accepting modifying text (selecting an alternative reading) for the sake of getting a unique referent. And in the previous section we saw another case where presuppositions could help us to choose which of two utterance readings to produce where a conjunction might be in the scope of a presuppositional environment (example (10)).[12]

Now that I have made my notion of presupposition clearer and have related it to the work described in previous chapters on definite NP processing, it should be apparent why incremental NLP systems are well-suited to computing presuppositions. Presuppositions are preconditions on the successful comprehension of particular utterance readings. When satisfied, they retrieve entities from the discourse model; these entities can be plugged into the reading being produced. When not satisfied, they can force the system to consider an alternative analysis whose presuppositions might be satisfiable.

Given these facts, it should be clear that an approach in which presupposition triggering and attempted satisfaction is an integral part of *utterance* processing is quite natural. Something like T42, where we check the acceptability of a syntactic and semantic analysis by interleaving presupposition checking with other processing, works well. We have also seen that the idea of "given" environments, effectively invoked by presupposition triggers, gives a natural account of the occasions when indefinite NPs appear to demand discourse model entities, as if they were definite NPs. This is explained by noting that in these cases these indefinite NPs occur within "given" environments. Again an incremental system which can have the notion of an "environment" embracing parts of an utterance can capture this easily. (It is also another factor in favour of treating presuppositions as preconditions

[11] In fact, in that system "presuppositions" are nothing more than selection restrictions (e.g. that the German verb "essen" means 'to eat' for human subjects; the German verb "fressen" means 'to eat' for animal subjects), which I do not regard as presuppositional.

[12] Of course, I am not denying that, in certain practical applications systems, treating presuppositions as ordinary inferences might still be best, given other factors that might influence the way processing is done, e.g. the cost of database access.

as opposed to their being ordinary inferences: with ordinary inferences we would not be able to show that these indefinite NPs should be regarded as referring expressions).

We will also see that the fact that T42 is incremental and works through an utterance from left-to-right is crucial to its approach to the projection problem for presuppositions (see Chapter 9), as its solution depends on retaining some notion of the order of occurrence of presupposition triggers.

7.5 Satisfying Presuppositions

In this section I look at the problems that arise in trying to satisfy a presupposition. While the examples concentrate on definite NPs, the problems apply to all presupposition sources. First I indicate that satisfying a presupposition may not be simply a matter of searching through the discourse model: entities may need to be inferred as well. Then I look at the reasons why the search for an entity might fail. As part of this, I question the assumption, implicit in previous parts of this thesis, that failure to find an entity (or infer one) will result in the utterance reading that is under construction being thrown away. I consider whether an entity should, in fact, be created instead. I find the evidence inconclusive, but I report a simple experimental change made to T42 that allows experimentation with this idea.

7.5.1 Inferring Discourse Model Entities

In referential uses of definite NPs and presuppositions in general, the speaker, in using a definite NP or other presupposition trigger, signals that the hearer should try to identify a referent; she expects the hearer to be able to find one. In T42 this is the case of solving a Constraint Satisfaction Problem using entities from the discourse model. I have perhaps given the impression up to now that the constraint satisfier simply retrieves entities directly from the discourse model. In this section I look at one of the problems that might arise with this: where an entity has to be inferred. These are cases in which the referent has been described in the literature as "inferable" (e.g. [Prince 1981], [Brown & Yule 1983]), and the process of inferring it as "accommodation" (e.g. [Lewis 1979], [Roberts 1987]). Hobbs has also considered such cases, with a treatment based on implicatures [Hobbs 1987]. They are cases where an entity satisfying the description cannot be found immediately but can be inferred through its relationship with other entities that have been mentioned in the discourse.

As an example, suppose we know that all cars have a steering wheel $\forall x$ $(\texttt{CAR}(x) \supset \exists y\ (\texttt{STEERING-WHEEL}(y) \wedge \texttt{HAS}(x,y)))$ and we have been talking about a car, then a definite NP can be used to refer to its steering wheel:

(13) *"Parolles has just bought a flash American car. The only trouble is that the steering wheel is on the left."*

T42 has been modified to handle simple examples of this. The axiom that cars have steering wheels is put into the knowledge base (world knowledge) in the form of a forward-chaining rule. For any entity in the immediate linguistic context (ILC) which gets the property `CAR` newly ascribed to it, the rule is triggered and for each car this creates a new entity that is labelled as the steering wheel of that car,[13] which is put into the ILC alongside the car that evoked it. Now the ILC will contain a steering wheel that can be felicitously referred to without being explicitly introduced using an indefinite NP. Of course, in practice, it would not be feasible to use simple forward inference in this way for more realistic knowledge bases: far too many entities associated with a car, say, would need to be added to the ILC just in case they are subsequently referred to (and knowing how to delimit this set of entities might be very difficult). However, the simple approach described is sufficient to allow me to develop presupposition handling in T42.

There are further difficulties of another kind though. These are cases where to infer an entity one needs to use *plausible* inferences as well as or instead of deductive ones. For example, the acceptability of (14a) and (14b) will depend on how plausible it is for you that picnics contain lemonade or beer respectively. If you can infer lemonade or beer on hearing of a picnic then entities can be retrieved for these. If not then utterance readings might not come off:

(14) a. "*Parolles took the picnic from the car. The lemonade was warm.*"
b. "*Parolles took the picnic from the car. The beer was warm.*"

T42 is not presently equipped with a plausible reasoning mechanism. It infers entities only using deduction.

7.5.2 Failure to Find an Entity

Below I list some of the main possible reasons why an ideal system might fail to find an entity when looking for one in a "given" environment, with different remedies for each (failure to find an entity in a "new" environment, of course, does not matter as an entity will be created anyway):

1. We might fail to find an entity in cases of lexical ambiguity. For example, in an utterance containing the word "bank", we might initially take "bank" to mean `RIVER-BANK`. Then, on consulting the discourse model, we find no river banks. For this situation, T42 would have the alternative analysis on an agenda of analyses that it has yet to try. This alternative, that "bank" means `FINANCIAL-BANK`, can be restored and tried.

2. We might fail to find an entity because we were treating the definite NP as being used referentially while in fact a non-referential use (i.e. an attributive or generic use) was intended. I have already pointed out that it is not necessarily the case that failure of a referential construal

[13] To get the rule to create entities, the rule has some procedural elements in it.

should give rise to a generic or attributive construal (Section 5.2). T42 cannot handle these other uses of NPs.

3. We might fail to find an entity if not enough inferencing was done: a bit more inferencing might yield a referent. There are a number of reasons why a line of inferencing might have been abandoned before it had had a full chance to see whether it could yield an entity. For example, the system might originally have considered some plausible inference as simply not plausible enough, or a line of inferencing (plausible or deductive) might have become so long that it was abandoned (i.e. so much work was involved in inferencing that this itself made the whole thing less plausible), or it might have been decided that not too much inferencing work would be done because the speaker was considered to be of low credibility, and so on. Unrealistically, T42 circumvents this problem by doing as much deductive inferencing as it can to find a referent: its search is exhaustive. If T42 did plausible inferencing or had a more realistically sized knowledge base, exhaustive search would be impractical and this problem would need solving.

4. We might fail to find an entity if we have considered only "salient" parts of the discourse model and if our notion of "salience" is too restrictive. In this case the search needs to spread further. Again T42 ignores this problem by unrealistically doing exhaustive searches.

5. We might fail to find an entity if the reading is simply contextually unacceptable. The speaker has referred to an entity for which the hearer has no counterpart in his discourse model. If all else fails, this is the conclusion T42 comes to. It then abandons the analysis.

The last of these might not be correct. A question that needs answering is: can a definite reference or other presupposition ever fail altogether?

Implicit in what I have been saying up to now is that if a presupposition cannot be satisfied by an existing discourse model entity (i.e. one cannot be found or inferred), and all other readings of the utterance also fail (e.g. readings for other senses of ambiguous words), then the utterance is wholly unacceptable. It fails completely. I will remind you that, in saying that there must be a suitable discourse model entity which can be picked up by the presupposition, I am not insisting that all the properties of the "given" environment already be in the discourse model. Chapter 6 showed that additional properties can also be ascribed in such environments. Nevertheless, so far I have insisted that an entity be *found* or *inferred*.

But it is possible that an entity should, in fact, be *created* instead. In the next subsection, I describe a variant of T42 that is able to do just that, i.e. as a last resort, it will create an entity to satisfy a presupposition rather than say that the utterance is wholly unacceptable. While the changes to T42 reported in that section demonstrate that this can easily be done within the general T42 framework, I do not necessarily believe that it is correct to do this. In order to discuss this issue, I first present some data which will be relevant:

(15) *"Parolles has just bought a flash American car. The only trouble is that the steering wheel is on the left."*

(16) *"Parolles took the picnic from the car. The beer was warm."*

(17) *"As soon as he turned the corner, he looked for the House above the Arch."*
(The first line of the novel *Chatterton* by Peter Ackroyd).

(18) A: *"Falstaff will be there."*
B: *"Do I know him?"*

(19) *"Brutus went to the bank."*

(20) *"If Macbeth bought a book, he'll be home reading it by now.*
**It/The book is a murder mystery."*

Example (15) we have already seen and I have suggested that an entity can be inferred using deduction over world knowledge. I have also suggested that example (16) requires a line of plausible inferencing to be established from the picnic to the beer. Example (17) shows a typical device used in the opening sequences of novels: definite NPs and perhaps other presupposition triggers are used even though the entities they refer to have not been explicitly introduced by, e.g., an indefinite NP. This device is used to give to the reader the idea that the character and, in (17), his quest are already familiar. We presumably have to create entities for the referents of the underlined NPs. Example (18) also seems to require that B create a discourse model entity to represent Falstaff in order to ask for further information about him. In (19) we must be careful: if we fail to find a river bank in the discourse model, this does not necessarily license the creation of such an entity. The discourse model may contain a financial bank, for example; on this reading, the presupposition is straightforwardly satisfied. Finally in (20) we have a case where the underlined definite NP cannot pick up the book being talked about in the conditional sentence. The explanation of this is given in Chapter 9.[14] I think (20), in the absence of any other book in the discourse model that it might refer to, is unacceptable. It would not be right to create an entity here.

Having looked at some of the data we can review some of the possible solutions to this problem. If we continue to do as T42 does, i.e. fail if an entity cannot be found or deductively inferred, then we will not be able to handle examples such as (17). If on the other hand we always create entities if they cannot be found, as the T42 variant described in the next section does, then examples such as (20) will be incorrectly found acceptable. The T42

[14] See also [Roberts 1987]. Roberts investigates, under the label of "modal subordination", cases where the definite could pick up the entity introduced in the conditional, as in:

> *"If Macbeth bought a book, he'll be home reading it by now.*
> *It/The book will be a murder mystery."*

variant is careful enough in cases such as (19), where there is the possibility of an alternative reading coming off: it creates entities only as a last resort.

Gazdar [1979] introduced the idea that consistency with context might be enough: as long as a presupposition is not directly contradicted by the context, it cannot fail. Mercer [1987], for example, has accepted this point of view (see Chapter 10). The problem with this comes with cases such as (19). Gazdar cannot exploit presupposition failure as I do to help choose readings for ambiguous utterances. If (19) is uttered in a context containing a financial bank, `F`, that has been under discussion but not containing anything to suggest that we might be talking about river banks, Gazdar's theory presumably predicts that this has two equally acceptable readings: in one we pick up `F`, the financial bank, as the referent of the definite NP; in the other, due to a lack of anything to the contrary, we are able to create and accept into the discourse model a new entity, `R` say, and incorporate into the context `RIVER-BANK(R)`. Some other mechanism must then choose between these two readings. T42, on the other hand, would pick up `F` to give one reading, but the lack of plausibility of river banks in the conversation means that we do not get another reading. Presuppositions have thus been the mechanism by which this utterance has been disambiguated. Similar examples have been given for PP attachment and determining the extent of conjuncts in presuppositional environments. Even the T42 variant described in the next section only creates entities as a last resort, when all readings have failed.

We have seen weaknesses with the approach that always fails, with the approach that always creates and with the approach that creates only if consistent. I believe that the correct approach is, as a last resort, when all other readings have failed, to create an entity if it is *plausible*. This, unfortunately, is very vague, but further investigation of it is beyond the scope of this dissertation.

7.5.3 Creating an Entity in a "Given" Environment

In this section I report the experimental change made to T42 that allows it to create entities if the demand for an entity in a "given" environment is not satisfied, rather than treating the utterance as unacceptable. The previous section discusses why this is a too liberal approach.

I equipped T42 with an extra 'register' to supplement the agenda (the agenda being where other possible analyses are held while the current analysis is processed further). As soon as a `GIVEN` constraint is encountered, a copy of the system state is put into this register, except that the `GIVEN` constraint is deleted from this copy. Processing proceeds as normal.[15] If the NP is successfully processed, on its closure the register would be emptied: since we managed to find a referent for the definite NP, we do not need to backtrack to the state that we stored in this register. If the CSP is unsatisfiable (a

[15] If another `GIVEN` constraint is encountered, due to there being a definite NP within another definite NP, for example, on seeing that the register already holds an earlier system state, nothing is copied to the register: this is so that if we are to backtrack to the state held in this register we go back to the first possible source of the problem.

satisfaction set becomes empty) then that reading does not come off and an alternative is taken from the agenda, this being an alternative analysis. Only if all possibilities from the agenda are tried and fail to come off is the register consulted. Then its contents are restored. So, for example, the system must first fail to find both a river bank and a financial bank before it restores the state from the register. What it will restore will not have the GIVEN constraint in it. Processing will restart as if the definite NP were like an indefinite NP in a "new" environment: it will both look up an entity (which will fail) and create a new entity (as only one entity is created and no existing entities will be found, the uniqueness constraint will be satisfied). In the example, it will get two readings, one for the river bank and one for the financial bank (and it has no mechanism for choosing between them).

We have already seen the problems this has. There are cases where it is a too generous approach and cases where it seems necessary: the truth is somewhere in between as I have said, with plausible reasoning being very important.

::::

This chapter has characterised my notion of presuppositions: they are preconditions, triggered by certain parts of speech, which demand that discourse model entities be found to satisfy them. Satisfaction is defined against the hearer's discourse model. I have shown that this approach is easily extended from definite NP processing to other presupposition triggers such as factive verbs. I have also contrasted this viewpoint of presuppositions with other approaches, in particular those taking presuppositions as ordinary inferences. These approaches do not really know what to do with presuppositions once they have computed them. I have also been able to handle cases such as those where an indefinite NP behaves anaphorically, requiring a discourse model entity. These are accounted for in T42 by noting that they occur in "given" environments. Finally I examined the notion of presupposition satisfaction further. I indicated that searching the discourse model alone might not be sufficient: some inferencing might be needed. I also considered what should happen in cases of presupposition failure: there are so many reasons why a presupposition might fail that it would be wrong to simply allow an entity to be created. However, in certain cases this does seem to be allowed *as a last resort*. I described a T42 variant in which I added this facility, but, as I said, acceptability should really depend more on plausibility. In the next chapter I examine how to account for presuppositional behaviour in utterances of negative sentences.

Chapter 8

Presupposition and Negation

In this chapter, I cover some of the most difficult data that affects presuppositions: their behaviour in utterances of negative sentences, including cases of so-called presupposition "cancellation". After a brief review of the problem, I have brought together all the approaches that have been given in the literature. I believe that there are five main approaches. For each, I describe their account of the data, indicate the general problems of the account and state why T42 cannot or does not use the approach in question. I conclude with a proposal for an account which is compatible with T42. I treat negation as presupposition preserving and handle presupposition "cancellation" metalinguistically. In this T42 is similar to the last of the five accounts, but the notion of metalinguistic I use seems different from the one used there.

8.1 The Problem of Negation

Presuppositions have been said to be constant under negation. That is, utterances of a positive sentence and its corresponding negative sentence will share the same presuppositions. This is in contrast to ordinary entailments which may fall under the scope of a negation operator. Both (1a) and (1b) license the presupposition (1c), i.e. in my terms, (1c) is a precondition on the hearer understanding readings for (1a) and (1b). Similarly, both (2a) and (2b) license (2c):

(1) a. "*The Prince of Denmark smiled.*"
b. "*The Prince of Denmark didn't smile.*"
c. 'There is a Prince of Denmark.'

(2) a. "*Lear regrets that he hit the fool.*"
b. "*Lear doesn't regret that he hit the fool.*"
c. 'Lear hit the fool.'[1]

This constancy under negation cannot be straightforwardly definitional, because it has been claimed that there are cases where it does not hold.

[1] There are, of course, other presuppositions of (2a) and (2b): that there is a Lear and that there is a fool. As has been my practice up to now, I continue only to show those presuppositions which I want to discuss.

These are referred to as cases of presupposition "cancellation". I will use this word with scare quotes throughout this chapter, because as far as my theory goes, it is a complete misnomer. For example, (3) allegedly does not presuppose (1c) and (4) allegedly does not presuppose (2c):

(3) "*The Prince of Denmark didn't smile because there isn't a Prince of Denmark.*"

(4) "*Lear doesn't regret that he hit the fool because he didn't hit the fool.*"

These are obviously "marked" or "dispreferred" uses of negation but nonetheless there are situations in which they would be acceptable. (In fact I believe that considering what these contexts are will be crucial to any solution to this problem).

In utterances of positive sentences, however, presupposition "cancellation" is allegedly not possible (without causing infelicity), as can be seen by conjoining the denial of the presupposition to the positive sentence:

(5) *"*The Prince of Denmark smiled but there is no Prince of Denmark.*"

(6) *"*Lear regrets that he hit the fool but he didn't hit the fool.*"[2]

There is thus an asymmetry between positive and negative utterances which must be accounted for.

I mentioned in Chapter 7 that the notion of presupposition "cancellation" also raises the question of how much later in a text a presupposition of a previous utterance can be felicitously cancelled, particularly given that presuppositions are quickly accepted as uncontroversial information.

Before I can explain my approach, it is necessary to look at various accounts of negation. There have been five main accounts:

1. Scope ambiguities for negation (Section 8.2 "Scope Ambiguities")
2. Different semantic operators for negation (Section 8.3 "Semantically Ambiguous Negation and Logical Presupposition")
3. Different pragmatic operators for negation (Section 8.4 "Pragmatically Ambiguous Negation Operators")
4. Vague negation (Section 8.5 "Vague Negation")
5. Ambiguity in the use of negation (Section 8.6 "Semantically Unambiguous Negation with a Metalinguistic Use)

As just indicated, the following subsections of this chapter review each of these in turn; Section 8.7 then presents T42's approach to negation.

[2] If you find this acceptable, a possible explanation is that you do not agree that "regret" has presuppositional behaviour. Try substituting "be aware" in its place.

8.2 Scope Ambiguities

Russell [1905 (1975)] has tried to account for the data using an ambiguity in the scope of negation. He can easily capture presupposition "cancellation" through differences in the scope of negation, i.e. differences in which of the conjuncts of a proposition (sentence reading) are negated. In particular, the presupposition-preserving reading uses narrow-scope negation. Thus this reading of (1b) ("The Prince of Denmark didn't smile.") will be:

$$\exists x\ (\texttt{PRINCE-OF-DENMARK}(x) \wedge\neg(\exists y\ (\texttt{SMILED}(y) \wedge \texttt{AGT}(y,x))))$$[3]

Wide-scope negation would give a presupposition-cancelling reading. For this reading of (1b) we get:

$$\neg(\exists x\ (\texttt{PRINCE-OF-DENMARK}(x) \wedge\exists y\ \texttt{SMILED}(y) \wedge \texttt{AGT}(y,x)))$$

As an account of the ambiguity of negative *sentences* Russell's account stands, but there is a general problem with his account. As Russell does not deal with *utterances*, he cannot explain why the narrow-scope reading is preferred and the wide-scope reading is marked. On top of this, the more specific reason why this account cannot be used within T42 is that I am not using Russell's analysis of noun phrases as existentially quantified expressions. Since in T42 definite noun phrases and other presupposition triggers pick up discourse model entities and plug these into logical forms, definite NPs can never fall within the scope of negation (i.e. they have wider scope than wide-scope negation).

8.3 Semantically Ambiguous Negation and Logical Presupposition

For my construal of Strawson [1952], the presupposition-preserving reading for (1b) is easy:

$$\texttt{PRINCE-OF-DENMARK(Hamlet)} \wedge\neg(\exists y(\texttt{SMILED}(y) \wedge \texttt{AGT}(y,\ \texttt{Hamlet})))$$

PRINCE-OF-DENMARK(Hamlet) cannot fall within the scope of negation since it must be true in order for us to be able to get the reading. But this means that Strawson cannot get the presupposition-cancelling reading.

Strawson is usually taken to have overlooked the problem of presupposition "cancellation". As far as I am aware, there is no explanation of such cases in his work [Strawson 1950, 1952, 1954, 1963].[4]

[3] Ignoring the uniqueness constraint.

[4] Kempson attributes to Bar-Hillel the ironic discovery that the final sentence of Strawson's "On Referring" is of a presupposition-cancelling form: "Neither Aristotelian nor Russellian rules give the exact logic of an expression of ordinary logic; for ordinary language has no exact logic." [Strawson 1950, p.344]

Keenan attempted to capture Strawson's notion of presupposition using the notion of logical or semantic presupposition [Keenan 1971]. In his account, presuppositions are propositions related to other propositions that represent sentence or utterance readings (it is not clear which):

> Proposition p presupposes proposition q iff p entails q and $\neg p$ entails q.

A well-recognised problem with this arises if a simple bivalent semantics is used (see, e.g., [Gazdar 1979, p.90]). Suppose the presupposition q is false (i.e. $\neg q$ is true). Since p entails q we can conclude that p must be false also. But if p is false then $\neg p$ is true and from $\neg p$ we can draw the presupposition q. Thus we have managed to derive $q \wedge \neg q$. Since this is a contradiction, our assumption that q was false can never hold: only tautologies can ever be presupposed, and this is clearly counter-intuitive.

To avoid this, logicians have proposed various non-bivalent semantics, e.g. the use of a third truth-value. Falsity of q then does not allow you to conclude that p is false: it can be false or have the third truth-value, i.e. presupposition failure leads to p having the third truth-value. I have already questioned whether this was what Strawson intended (Section 5.3).

The problem with this approach is that its *definition* of presupposition embodies the idea of constancy under negation. How is it then to account for cases of presupposition "cancellation"? The solution is to make natural language negation ambiguous. Negation in a natural language utterance will be translated by one of two operators: an *internal* negation operator or an *external* negation operator. The internal operator is like Russell's narrow-scope negation and preserves presuppositions; the external operator is like Russell's wide-scope negation and does not preserve presuppositions. Since logical presupposition demands a non-bivalent semantics, these operators are also non-bivalent. (We could not have two different negation operators if the semantics were not non-bivalent). There are various arguments against this approach:

- Gazdar's main counter-argument [Gazdar 1979, pp.65-66] is that he knows of no language where "not" has two lexical realisations. If it *is* ambiguous one might expect some lexical or syntactic evidence in some language or another.

- If a language incorporates two negations, then both would be normal senses (neither one would be "marked" or "dispreferred"). If both are perfectly normal then they could not account for the markedness of presupposition-cancelling negation [Burton-Roberts 1989].

- We need to modify the definition of logical presupposition to take into account the two negations: p presupposes q if p and the presupposition-preserving negation of p entail q. This is clearly vacuous [Burton-Roberts 1989]: the definition of presupposition depends on the definition of presupposition-preserving negation, which itself is defined only in terms of presuppositions.

T42 could use this approach but I believe that the general problems above are enough to militate against using it.

8.4 Pragmatically Ambiguous Negation Operators

Karttunen & Peters (K&P) [K&P 1979] are associated with a particular approach to determining the presuppositions of complex sentences. Their solution to this is described more fully in Chapter 10, but I will bring out the points that are relevant to negation here.

In K&P's solution, the various embedding constructs such as verbs of propositional attitude, negation and the connectives are classified as either "plugs", "holes" or "filters". Briefly, plugs are constructions that never let the presuppositions of their embedded complements become presuppositions of the utterance as a whole, holes always allow embedded presuppositions to surface, and filters sometimes do and sometimes do not.

K&P have two types of negation operator: an "internal negation", which is a hole, and an "external negation", which is a plug. Thus, internal negation allows presuppositions to survive; external negation does not and so functions as a presupposition-cancelling negation.

While this theory treats negation as ambiguous, there is something unusual in this. Ordinarily, internal and external negation, as described in the previous section, are distinguished by having different truth-tables: this can only really be so in a non-bivalent semantics. Yet K&P use a bivalent semantics. Their notions of internal and external negation cannot be the traditional ones. K&P suggest that negation is *pragmatically* ambiguous, rather than semantically ambiguous, i.e. it has more than one *use*: a use as a plug and a use as a hole.

All three of the problems of semantically ambiguous negation probably apply to K&P's pragmatically ambiguous negation. But, it is not at all clear what it means to have a pragmatically ambiguous operator. No definition is forthcoming from K&P. In Section 8.7 I give one characterisation of pragmatic ambiguity, and this does not tally with K&P's use of the term. Again evidence is against this approach and so T42 does not use it.

8.5 Vague Negation

Kempson [1975, pp.11-16] claimed that negation is vague, not ambiguous. Most treatments of presuppositions since then have accepted this (e.g. [Gazdar 1979], [Mercer 1987]). In this conception, negation does not have different semantic representations with distinct truth-conditions. It has one representation with disjunctive truth-conditions: "An ambiguous sentence is formulated as having two quite separate structures, whereas a vague sentence is one which is characterised semantically as a disjunction." [Kempson 1975, p.16]. The disjunction says that there are a number of ways of interpreting the single representation to make it true.

Kempson gives an analogy using the word "neighbour". This is vague, i.e. unspecified, as to the gender of the person amongst many other things (race, size, hair colour, ...). If negative sentences are to be ambiguous simply because they have disjunct readings, then likewise "neighbour" should be lexically ambiguous between a male neighbour and a female neighbour. This would be counter-intuitive.

Kempson gives a test for distinguishing ambiguity from vagueness: "If a sentence is ambiguous, then in order for verb-phrase pronominalisation to take place in a conjoined structure containing that sentence, the two conjuncts must agree in their interpretation of the ambiguous sentence." [*ibid.*, p.15].[5] The test tells us that the phrase "visiting relatives" is ambiguous rather than vague: it can mean going to visit relatives or it can mean relatives who visit. The evidence for this is that in VP pronominalisation, the conjuncts must agree on their interpretation: crossed interpretations, i.e. continuations which suggest they do not, as in (7), are anomalous:

(7) ??"*Casca likes visiting relatives and Brutus does too. Casca likes going to see relatives and Brutus likes them to come and see him.*"

However "music" is vague between pop, classical and jazz, etc.: the following text is felicitous even though the conjuncts differ in their interpretations:

(8) "*Casca likes music and Brutus does too. Casca likes pop and Brutus classical.*"

The test shows that negation is vague: the following text is felicitous, where the interpretation of the negation differs in the conjuncts:

(9) "*Casca didn't run away and Brutus didn't either. Casca walked slowly off and Brutus stayed stock still.*"

In theories that use vague negation it has been represented semantically by wide-scope negation. Thus (10a) would be represented by (10b):

(10) a. "*The Prince of Denmark didn't smile.*"
b. $\neg(\exists x \exists y(\texttt{PRINCE-OF-DENMARK}(x) \wedge \texttt{SMILED}(y) \wedge \texttt{AGT}(y,x)))$

This shows that this negative can be true if one or both of the conjuncts is false. These theories then need a pragmatic account of why it is that in normal circumstances one interpretation of (10b) is preferred over the others, i.e. the preferred interpretation is the narrow-scope one:

$$\exists x\ (\texttt{PRINCE-OF-DENMARK}(x) \wedge \neg\exists y\ (\texttt{SMILED}(y) \wedge \texttt{AGT}(y,x)))$$

[5] The test has problems [Kempson 1979]: cases of ambiguity where the more specific reading of the sentence entails the more general reading will allow crossed interpretations.

The preferred reading entails the "presupposition". If one can explain why this reading, which entails the "presupposition", is preferred then the notion of presupposition can be disposed of: to the extent that presuppositions exist, they will simply be entailments of positive sentences and entailments of the preferred readings of negative sentences.

Alternatively, in contexts where we know $\neg\exists x$ $(\texttt{PRINCE-OF-DENMARK}(x))$, then the only interpretation of (10b) that the context would be able to support is $\neg(\exists x \exists y(\texttt{PRINCE-OF-DENMARK}(x) \wedge \texttt{SMILED}(y) \wedge \texttt{AGT}(y,x)))$, which no longer entails the "presupposition". This is the case of presupposition-cancellation.

Theories based on vague negation have been proposed in [Kempson 1975], [Wilson 1975], [Gazdar 1979], [Atlas & Levinson 1981] and [Mercer 1987]. They all offer slightly different accounts of the preferred reading and in certain cases (Gazdar and Mercer) retain some notion of presupposition (at least for negative sentences).

The following subsections review some of these alternative theories to see what accounts they give of preferred readings and of the 'background feel' of "presuppositions" (given that presupposition as a strict notion has been disposed of and is now simply accounted for using entailment).

8.5.1 Wilson

Wilson's account [1975] of preferred interpretations runs as follows. Consider the utterance (11a), which, given that vague negation is represented as wide-scope negation, can be paraphrased as (11b):

(11) a. "*The Prince of Denmark didn't smile.*"
b. 'It is not the case that the Prince of Denmark smiled',
i.e. $\neg(\exists x \exists y$ $(\texttt{PRINCE-OF-DENMARK}(x) \wedge \texttt{SMILED}(y) \wedge \texttt{AGT}(y,x)))$

The possible interpretations that make this true are:

(12) a. 'The Prince of Denmark does not exist',
i.e. $\neg(\exists x$ $(\texttt{PRINCE-OF-DENMARK}(x)))$
b. 'The Prince of Denmark exists but didn't smile',
i.e. $\exists x$ $(\texttt{PRINCE-OF-DENMARK}(x) \wedge\neg(\exists y$ $(\texttt{SMILED}(y) \wedge \texttt{AGT}(y,x))))$

(12a) and (12b) are both possible interpretations but (12b) is preferred. Wilson's explanation of this preference is as follows. One of the strictures of Grice's Maxim of Manner is to be brief. It is reasoned that to use (11a) to convey (12a) would violate this maxim: there are briefer (or, at least, more direct) ways to convey (12a), e.g. by saying "There's no Prince of Denmark.". Hence (12b) is the preferred reading and this entails $\exists x$ $(\texttt{PRINCE-OF-DENMARK}(x))$. So a simple Gricean argument determines the preferred interpretation. Of course, context might rule this reading out in a particular case and, since another reading would then be selected, the entailment (presupposition) would be lost.

8.5.2 Atlas & Levinson

Atlas & Levinson (A&L) [1981] have a similar account to Wilson's. They use an extended Gricean argument to explain why the the narrow-scope reading is the preferred one. Their argument is slightly different from Wilson's because they claim that it is not clear that Grice's maxims do work as Wilson says they do.

A&L say that the narrow-scope reading is given by a generalised conversational implicature. The generalised conversational implicature is one which implicates that the most informative interpretation of the utterance should be preferred. However, A&L say that this cannot be derived from Grice's maxims as originally stated and, in fact, it is possible to derive the opposite conclusion. I will illustrate this with an example not involving negation first. (13a) supposedly implicates the more specific (13b):

(13) a. "*Richard III jumped on his horse and rode into the sunset.*"
b. 'First Richard III jumped on his horse and then he rode into the sunset.'

But A&L claim that (13b) cannot in fact be derived from (13a) by Gricean maxims. Indeed, the maxim of Quantity would lead us to conclude that, since the speaker chose to say (13a) instead of uttering the stronger statement (14a), far from implicating (13b), she implicates that she is *not* in a position to convey (13b) and hence implicates its opposite, (14b):

(14) a. "*First Richard III jumped on his horse and then he rode into the sunset.*"
b. 'Richard III jumped on his horse and rode into the sunset but not in that order.'

To remedy this, A&L introduce a new principle:

> **Principle of Informativeness**
> The 'best' interpretation of an utterance is the most informative one consistent with what is non-controversial. [Levinson 1987, p.66]

This now accounts for the data since "[v]ery often the 'best interpretation' of what is said will be a more specific interpretation in line with *stereotypical expectations.*" [*ibid.*, p.66][6]

Thus A&L are claiming that without their new principle, Wilson is wrong in assuming that Grice's theory will license the more specific reading. But

[6] This does introduce a problem: given that the maxim of Quantity and A&L's Principle of Informativeness give conflicting implicatures, which of them will prevail in any one particular case? Assessing A&L's resolution of this is beyond the scope of this dissertation. They basically say that Quantity implicatures are favoured over Informativeness implicatures as long as this is consistent with background knowledge. Often it will not be consistent with background knowledge: the Informativeness implicature, which fits in with stereotypical expectations, is more likely to be consistent with background knowledge and so likely to survive more often.

with their principle the presupposition preserving reading is licensed. They are probably right about this, but Grice's theory is vague enough to be consistent with a number of interpretations. Which account is right does not really matter; all that counts is that a Gricean or neo-Gricean account is used to determine the preferred reading.

8.5.3 Kempson

Kempson too [1975, Chapter 5] says that "presuppositions" are simply entailments of the positive sentence and entailments of one of the readings of the negative sentence, and so she too needs to appeal to a preferred interpretation of negatives.

She does not follow Wilson and A&L, however, i.e. she does not say that the presupposition-preserving reading is conversationally implicated. Rather she says that these entailments ("presuppositions") are not just entailed, they are also conversationally implicated to be true (in both positive and negative utterances). We then prefer a reading which preserves as many of these conversational implicatures as possible. This will obviously be the narrow-scope, presupposition-preserving reading. Since conversational implicatures can be "cancelled", we will be able to get the presupposition-cancelling reading: in these the implicatures will not hold and so the wide-scope reading, which does not have the entailments, will be indicated.

Kempson needs to account for why it is that certain entailments (the ones traditionally taken to be "presuppositions") are also conversationally implicated. Her account [Kempson 1975, pp.173-185] can be abbreviated as follows. Definite NPs with textual antecedents cannot fall within the scope of negation. For example:

(15) *"*The Prince of Denmark smiled but he didn't laugh because there is no Prince of Denmark.*"

The speaker's first conjunct in (15) entails the existence of the Prince of Denmark. In the rest of (15), the speaker then denies the existence of the Prince of Denmark. Clearly this makes the utterance self-contradictory. In cases where a presupposition could be cancelled (where the antecedent is set up intersententially or extralinguistically), the speaker has implicated to the hearer that she expects the hearer to be able to identify a referent: she has done this by choosing to use the definite article, which usually serves this role with textual antecedents, rather than by choosing to use the indefinite article.[7]

The nice thing about this account is that these generalised conversational

[7] Kempson accounts for other presuppositional phenomena such as factives by claiming that they have an underlying definite NP [Kempson 1975, pp.185-188]: "Lear regrets that he hit the fool" has "Lear regrets *the fact that he hit the fool.*" as its underlying form. It is not clear that this is correct or whether it easily extends to other cases, e.g. does "Lear stopped beating the fool" have "Lear stopped his/the beating of the fool" as its underlying form?

implicatures can be said to account for the background feel of "presuppositions" too.

8.5.4 Summary of Vague Negation

These accounts of vague negation can be summarised as follows.

Wilson, A&L: There are no presuppositions; generalised conversational implicatures determine a preferred reading for negatives; "presuppositions" are entailments of the preferred reading; however, this does not account for the background status of "presuppositions" in either negative or positive sentences.

Kempson: There are no presuppositions; "presuppositions" are generalised conversational implicatures; the preferred reading of a negative is the one which preserves these implicatures; these implicatures will then also be entailed by the positive and the preferred reading; moreover, the fact that these inferences are implicatures accounts for their background status in some sense.

The popularity of the vagueness approach to negation is testimony to the fact that it does seem able to account for the data, even though Wilson and A&L offer no account of the 'background feel' of presuppositions and Kempson's account may not extend easily to things other than definite NPs. More to the point, these theories seem to take the wide-scope presupposition "cancelling" reading as primary and then offer a special pragmatic account of why the narrow-scope presupposition preserving reading is preferred. More intuitive would be an account that takes the narrow-scope presupposition preserving reading as primary and offers a special pragmatic account of the presupposition "cancelling" reading [Burton-Roberts 1989].

But in any case, T42 cannot use the vague negation approach because T42 always satisfies presuppositions with discourse model entities, which effectively means that it takes presuppositions to have wider scope than negation; this is contrary to the representation of vague negation as having wider scope than presuppositions.

8.6 Semantically Unambiguous Negation with a Metalinguistic Use

Burton-Roberts [1989] says there is no semantic ambiguity of negation and an account in terms of vague negation is not needed: natural language negation is simply internal (presupposition-preserving) negation. However, for this to be so, a non-semantic, non-truth-conditional, pragmatic account of presupposition "cancellation" is needed. For this, Burton-Roberts looks to "metalinguistic negation" (see below). This is the approach which is most similar to T42's, inasmuch as both treat natural language negation as unambiguously narrow-scope and both treat presupposition "cancellation" as metalinguistic. However, the two notions of metalinguistic seem to be different.

Burton-Roberts says that vague negation theories (as described in the previous section) use a single, wide-scope negation and a pragmatic explanation of why narrow-scope negation is preferred, whereas he uses a single, narrow-scope negation and a pragmatic explanation of marked, dispreferred negation (presupposition "cancellation").

> "The theories [i.e. Burton-Roberts' theory and the vague negation theories] must be seen as being in agreement that negation is semantically unambiguous. The issue in contention is this: which understanding of negation is to be taken as the single truth-functional (logical) semantic negation and which pragmatically derived? Critics of truth gap theories are committed to its being the 'presupposition-cancelling' (external, so-called 'denial') negation that is semantically basic and that any other understanding of negation is to be derived from that semantic operator in pragmatic terms, in terms of an explanation of a particular use of that semantic operator in utterances ... Diametrically opposed to this, proponents of truth gap theories are committed to its being the 'presupposition-preserving' (internal, descriptive) negation that is semantically basic and that any other understanding of negation is to be derived pragmatically, in terms of an explanation of a particular use of that operator." [Burton-Roberts 1989, pp.112-113].

An immediate reason for preferring Burton-Roberts' account is that it treats the narrow-scoping as 'normal' and the wide-scoping as the one needing the special explanation ("marked").

Burton-Roberts uses "metalinguistic negation" to obtain the presupposition "cancelling" reading. The essential point about metalinguistic negation is that it is a *rejoinder* to a previous utterance, not a semantic denial. Metalinguistic negation does not assert the falsity (deny the truth) of an utterance reading but rejects an utterance as unassertable. It is "... a device for registering objection to a previous utterance (not proposition) on any grounds whatever, including the way it was pronounced." [Horn 1985, p.121].

The archetypical case of metalinguistic negation is (16a). Horn [1985] and Burton-Roberts [1989] would also say that (16b), (16c) and (16d) involve metalinguistic negation:

(16) a. "*It isn't a tom[a:]to, it's a tom[eɪ]to.*"
b. "*Some men aren't chauvinists, all men are.*"
c. "*Macbeth didn't meet two witches, he met three.*"
d. "*The Prince of Denmark didn't smile — there isn't a Prince of Denmark.*"

In these examples, the truth-conditional content does not fall under the negation: the speaker is *objecting to a previous utterance* on the grounds of its phonetic realisation (16a), its implicatures, (16b) and (16c), and its presupposition (16d). Metalinguistic negation is not logical negation but rather is a metalinguistic denial of the appropriateness of an utterance; for example, (16c) is really conveying something like:

> 'You cannot appropriately describe Macbeth as having met two witches, you should describe him as having met three.'

Burton-Roberts and Horn say that (16b) and (16c) must have metalinguistic readings since if they do not the implicatures, in order to be subject to normal logical negation, must be allowed to embellish the logical form. But this is contrary to what most people, including Burton-Roberts and Horn, think implicatures should do. (However, it is not out of the question for, as I pointed out in Chapter 2, this is precisely what Levinson's pragmatic intrusion account does allow [Levinson 1988]).

In Burton-Roberts' account, the first conjunct of (16d), using the narrow-scope reading of the negation, entails the presupposition (that there is a Prince of Denmark), but then the second conjunct asserts the opposite: this (as in all the examples) sets up a contradiction on initial analysis. In order therefore for this utterance to be interpreted as a coöperative one, the hearer is forced to reinterpret (16d) as involving a metalinguistic use of negation which will convey something that is non-contradictory (that the speaker objects to the form by which the proposition was conveyed rather than its propositional content). In this case, according to Burton-Roberts, the proposition will have no truth-value.

Metalinguistic negation, as a rejoinder to a previous utterance, is supposed to "echo" that utterance. For example, (16d) is supposed to be a rejoinder to a previous utterance of "*The Prince of Denmark smiled.*", as in (17). Burton-Roberts does not mention that (16d) could equally well follow an utterance of "*The Prince of Denmark didn't smile.*", as in (18).

(17) A: "*The Prince of Denmark smiled.*"
B: "*The Prince of Denmark didn't smile because there isn't a Prince of Denmark.*"

(18) A: "*The Prince of Denmark didn't smile.*"
B: "*The Prince of Denmark didn't smile because there isn't a Prince of Denmark.*"[8]

Burton-Roberts and Horn give support to the idea that metalinguistic uses "echo" previous utterances by showing that negative polarity items can only appear in a metalinguistic use if they can also appear in the original positive utterance which the metalinguistic use objects to. For example since (19a), which contains negative polarity items ("any" and "yet"), would not be a felicitous positive utterance, then (19b), which is a rejoinder to (19a), cannot contain the negative polarity items. Only if a negative polarity item can occur in the positive may its metalinguistic rejoinder contain

[8] One would seem to need different intonation patterns to make these felicitous: in B's utterance in (17) one puts focus on "did", in B's utterance in (18) the focus is on "smile".

the negative polarity item, even though ordinary semantic negation would normally demand the use of negative polarity items. But since I have shown that Burton-Roberts' metalinguistic uses can also follow negative utterances, which may contain negative polarity items, as in (20a), then metalinguistic uses may, in fact, contain negative polarity items, as in (20b):

(19) a. ??"*The Prince of Denmark has lost any hair yet.*"
b. ??"*The Prince of Denmark hasn't lost any hair yet; there's no Prince of Denmark.*"

(20) a. "*The Prince of Denmark hasn't lost any hair yet.*"
b. "*The Prince of Denmark hasn't lost any hair yet; there's no Prince of Denmark.*"

An apparent advantage of the metalinguistic approach is that it gives a uniform treatment of all the examples in (16). Each example is a case where on initial analysis the truth-conditions are self-contradictory. That they are all self-contradictory is perhaps the most important point about cases of metalinguistic negation. It is this self-contradiction that triggers for the hearer the realisation that a re-analysis in terms of metalinguistic negation is called for.[9]

But it would seem reasonable to question this claim to uniformity. While the pronunciation example (16a) would seem to require the notion of metalinguistic negation, Levinson can offer a different account for the cases involving conversational implicatures such as (16b) and (16c). He does not regard them as metalinguistic negation; he uses his "pragmatic intrusion" account instead ([Levinson 1988] — see Chapter 2). He says that the implicatures intrude into and thus embellish the logical form. Once this has happened, they are straightforwardly available to be negated, logically. Levinson prefers this account for three reasons: (a) there are other cases of intrusion (such as comparatives and conditionals); (b) there are propositions being denied in (16b) and (16c) above which is not the case in (16a); and (c) he says that present accounts of implicatures in negative utterances (e.g. [Gazdar 1979] and [Hirschberg 1985]) give wrong predictions: "Macbeth didn't meet two witches" does not implicate 'Macbeth met no more than two witches." Given that the latter is not implicated it obviously cannot be cancelled by metalinguistic negation (it isn't there to cancel)[10].

If Levinson is right that (16b) and (16c) are instances of a phenomenon different to that which accounts for (16a), then (16d) might have some other

[9] An exception, which is regarded as a case of metalinguistic negation but does not set up a contradiction, is:

"*The next Prime Minister won't be Wilson; it'll be Heath or Wilson.*"

From this we can straightforwardly deduce that the next P.M. will be Heath: there is no contradiction. Yet this utterance really demands a metalinguistic reading. Burton-Roberts says that this is accounted for by saying that the conclusion that it will be Heath is contradictory to "the obvious intention behind its utterance." [Burton-Roberts 1989, p.116-117].

[10] He says that "Macbeth didn't meet two witches" does not preserve the 'no more than two' implicature nor does it simply block it. You get different implicatures in negatives, in this case that Macbeth met at least one witch.

explanation too.[11] In the next section I give my account of cases such as (16d): I follow Burton-Roberts in taking negation to be semantically unambiguous, and while I accept that cases such as (16d) are metalinguistic, I do not resort to metalinguistic *negation* to handle them.

If we accept Levinson's account for (16b) and (16c), and assume that I can offer a different account for (16d), this leaves only (16a) as an actual case of metalinguistic negation in Burton-Roberts' and Horn's sense. Examples similar to (16a) would be "*I'm not his daughter — he's my father!*", "*The bottle isn't half empty, you pessimist, it's half full!*" and "*I didn't trap two mongeese, I trapped two mongooses!*". A treatment along the lines of metalinguistic negation seems quite appropriate for these: none of them are denying propositions, all of them are questioning the surface form of an utterance with the negation wrapped around a sort of quotation of (part of) a previous utterance. By contrast, the other examples, (16b), (16c) and (16d), do seem to be denying some proposition, rather than announcing the speaker's unwillingness to accept someone else's way of asserting something.

A further general problem with Burton-Roberts and Horn's account is that they have not really said what metalinguistic negation is. Horn describes it as an "ambiguity of use" and as a "pragmatic ambiguity". Even Burton-Roberts says that it is not clear what Horn means by these; instead, Burton-Roberts simply says that self-contradictory utterances force a re-analysis in terms of metalinguistic negation. How metalinguistic negation would be represented or how it could be determined exactly which part of a previous utterance is being objected to are not clear. In Section 8.7.2 I investigate what a pragmatic ambiguity is in the context of my discussion of how T42 handles presupposition "cancellation" and show there that Burton-Roberts and Horn's metalinguistic negation does not fit the notion of pragmatic ambiguity that is most consistent with the present literature.

This concludes my review of the various approaches to negation (scope ambiguity, semantic ambiguity, pragmatic ambiguity, vague negation and unambiguous negation with a metalinguistic use). I have indicated both the general problems of each and, where applicable, the problems that they cause for T42 in particular. The result is that T42 uses an approach that is similar, but not identical, to Burton-Roberts' account.

8.7 T42's Processing of Negation

For T42, negation is semantically unambiguous; it is internal, presupposition-preserving, narrow-scope negation. Presupposition "cancellation" is metalinguistic, a case of the speaker giving information about the model-theoretic interpretation of propositions in the discourse model. This metalinguistic behaviour is different to Burton-Roberts' metalinguistic negation, which is a rejoinder to a previous utterance.

[11] There is further support that (16d) might not be wholly uniform with the other examples, which is that in (16d) the metalinguistic use objects to presuppositions, which, in some theories, are part of truth-conditional content, while in the other examples something non-truth-conditional is being objected to.

I will now describe T42's processing of straightforward cases of negation and then bring out its handling of "cancellation" in a subsequent subsection.

8.7.1 T42 and Presupposition-Preserving Negation

For T42 I have only looked at simple uses of the word "not" where it is a sister to a verb phrase. Other uses of "not" (e.g. "It is not the case that ..."), other forms of negation (e.g. "never", "no one", etc.) and morphologically incorporated negation (e.g. "unhappy") have not been considered. In T42 the occurrence in a sentence of the word "not" sets a flag, NEG. The lexical entry for "not" is therefore as follows:

word	*category*	*meaning*
"not"	$(S_x\backslash NP_y)/(S_x\backslash NP_y)$	⋆NEG

While this flag is set, we are within the scope of negation. The word "not" in English usually signals the start of the scope of negation with clause boundaries marking the end.[12] We have already seen that in "new" environments T42 both looks entities up in the discourse model and creates new entities. This behaviour is modified when the GIVEN flag is set: new entities are not created, only existing ones can be used. The NEG flag has the opposite effect: *only new entities* are used; the discourse model is not consulted. Entities which are newly created while the NEG flag is set receive a special label, which is also called NEG. In later parts of the utterance, if we are searching for an entity, *entities labelled* NEG *are to be regarded as inaccessible*, i.e. they cannot be used as candidate referents. This is because they ultimately translate to existentially quantified variables within the scope of negation (see below).

The flag is unset when the end of the clause is encountered but may also be temporarily unset ("usurped") if the GIVEN flag is encountered and then reset when the "given" environment is complete. Thus the GIVEN flag has temporary priority over the NEG flag. Once NEG is unset, normal processing resumes: discourse model entities may once again be used as candidate referents. At the end of processing an utterance, we want to read a logical form out of the immediate linguistic context (ILC) for transfer to the non-immediate context (NIC). At this stage, any entity labelled NEG is not

[12] This is a simplification that I have made because it makes life easier in a left-to-right system such as T42. Unfortunately, the scope of negation can cover items that precede the word "not" and can also be discontinuous. Furthermore, once one has determined which items fall within the scope of negation, the negation is in fact vague as to which particular part of those items are being negated. We can see this by noting that the following utterance can be felicitously followed by any of (a)-(e):

> *"Lear didn't see a woman."*
> (a) *"— it was a man he saw."*
> (b) *"— it was a girl he saw."*
> (c) *"— it was a boy he saw."*
> (d) *"— it was a tree he saw."*
> (e) *"— he heard her."* etc.

The continuations show that different aspects of the act of seeing a woman can actually be negated: the sex, the age, the sex and age, etc.

translated as a constant but as an existentially quantified variable. Furthermore all predications on these variables will fall within the scope of a negation operator[13].

It is important to note that the use of entities labelled NEG does not introduce anything special into the ILC. Again this is simply an unusual data structure for holding formulas of logic. When the constraint satisfier consults the ILC, it will know that it cannot use these 'entities' to solve its CSP. When the inference engine consults the ILC (e.g. at the end of the sentence), it knows that the logic is stored unusually and so must be converted to clausal form. The model-theoretic interpretation does not change.

A simple example will show this at work:

(21) "*The Prince of Denmark didn't smile.*"

The subject NP is processed as normal: assuming that the discourse model contains PRINCE-OF-DENMARK(Hamlet), T42 retrieves the entity Hamlet from the discourse model. It then moves on to "did", which has no significant effect[14]. The next word is "not", which sets the NEG flag. Until this is unset, the discourse model will not be consulted. Therefore, in processing "smile", T42 does not search for any existing discourse model entities. Rather, it simply creates a new constant, EV0 say, and labels it NEG. Finally, the agent of "smile" can be equated with the subject NP. The NEG flag can then be unset. In reading the logical form out of the ILC, since EV0 is labelled NEG, it is translated as an existentially quantified variable within the scope of a negation operator:

$$\texttt{PRINCE-OF-DENMARK(Hamlet)} \wedge\neg(\exists ev_0\ (\texttt{SMILE}(ev_0) \wedge \texttt{AGT}(ev_0, \texttt{Hamlet})))$$

This is narrow-scope negation: there was no question of the subject NP falling within the scope of negation.

In the next example a presupposition trigger occurs after the "not" but processing is similar to the previous example: the presupposition must still be satisfied.

(22) "*The Prince of Denmark didn't choose the play.*"

I assume that PRINCE-OF-DENMARK(Hamlet) and PLAY(P) are in the discourse model. The subject NP is processed as it was in the previous example.

[13] This scheme lacks elegance but it mostly works! A problem case is "Lear regrets he didn't hit the fool." where a NEG occurs *within* a "given" environment. These cases are not presently handled. Extending the ontology of the logic so that discourse model entities can denote 'negative events' is not appealing. An alternative, suggested to me by Steve Pulman, (although I am not certain that this is greatly different and it might introduce problems of its own, not least since it involves using higher-order logic) is to extend the ontology to have discourse model entities that denote "facts", e.g. FACT(F1, SMILE(EV1) ∧ AGT(EV1, Hamlet)) says that F1 is the fact that Hamlet smiled, and equally FACT(F2, $\neg\exists ev_2$(SMILE(ev_2) ∧ AGT(ev_2, Hamlet))) says that F2 is the fact that Hamlet did not smile. It is then possible that factive verbs could presuppose the fact of their complements.

[14] I have not considered how to handle tense in T42.

The "not" again sets the NEG flag. A choosing event, EV1, is newly created (the discourse model is not consulted for any existing choosing events because the NEG flag is set) and is labelled NEG. Its agent is the subject NP. T42 then begins to process the object NP. On encountering ⋆GIVEN in the lexical entry for "the", the NEG flag is unset and the "given" environment takes effect: P is retrieved from the discourse model and made the object of EV1. The following is then read out of the ILC:

$$\texttt{PRINCE-OF-DENMARK(Hamlet)} \wedge\neg(\exists ev_1\ (\texttt{CHOOSE}(ev_1) \wedge \texttt{AGT}(ev_1,\ \texttt{Hamlet}) \wedge \texttt{PAT}(ev_1,\ \texttt{P}))) \wedge \texttt{PLAY(P)}$$

Again this is presupposition-preserving negation. Indeed, if T42 had failed to find referents for "the Prince of Denmark" or "the play", this reading would not have come off.[15]

Notice that the same reading would have been generated for "*The play wasn't chosen by the Prince of Denmark.*". In accord with my intuitions, the active and the passive have the same presuppositions. In some other accounts of negation, actives and passives might be assigned readings with different presuppositions.

The examples show that the presupposition triggers are always actioned and must always be satisfied. Anything within a "given" environment triggers a demand for discourse model entities as before (even in negative utterances). This would appear to give me an account like Strawsons's. Both positive and negative sentences trigger presuppositions and the discourse model must satisfy these presuppositions else the reading does not come off. But how then will I account for presupposition "cancellation"?

8.7.2 Presupposition "Cancellation" in T42

In this subsection I will be proposing that presupposition "cancelling" utterances are in fact metalinguistic utterances conveying 'discourse about discourse'. For Burton-Roberts and Horn (see Section 8.6), there is specifically a metalinguistic *negation* (or metalinguistic *use* of negation), which is used to deny, e.g., the surface form of a previous utterance: it is a sort of quotational use. I have agreed that an explanation of this kind seems necessary to account for certain utterances such as (16a) above. However, I have questioned whether this is what is at work in cases of presupposition "cancellation", and for these cases I use a different notion of "metalinguistic".

I want to propose not a metalinguistic (use of) *negation* but rather that certain *utterances* are metalinguistic. The analogy is with formal languages,

[15] You will recall that at the end of Chapter 7 I described a variant of T42 which, if a presupposition failed, as a last resort, would reprocess a "given" environment as if it were a "new" environment and hence be able to create new entities to satisfy failed presuppositions. If this T42 variant has to create a new entity as the referent of "the play", since it does this within the scope of negation, the entity will be labelled NEG and so incorrectly translated as an existentially quantified variable. The incorrect reading it gets will be: $\texttt{PRINCE-OF-DENMARK(Hamlet)} \wedge \neg(\exists ev_1 \exists p_1\ (\texttt{CHOOSE}(ev_1) \wedge \texttt{AGT}(ev_1,\ \texttt{Hamlet}) \wedge \texttt{PLAY}(p_1)))$. This is another reason for dispreferring this variant.

such as logic, where another language, such as English, is used as a metalanguage to describe properties of the interpretation of the formal language. The difference in the case of English (and other natural languages) is that both object language and metalanguage utterances will be in English. This might give us some trouble in distinguishing the two. Hence I will continue now by discussing how T42 spots a metalinguistic utterance: the problem of recognition. With this done, I will later discuss in more detail what a metalinguistic utterance conveys and what T42 should do having spotted one.

Consider the following cases of presupposition "cancellation":

(23) a. "*The Prince of Denmark didn't smile: there isn't a Prince of Denmark.*"
b. "*The Prince of Denmark didn't choose the play: there wasn't a play.*"

The first and most important thing to consider is when, i.e. in what contexts, the speaker would utter such sentences. They are not utterances that would come "out of the blue" (indeed, no utterance is, but these utterances demand a special context of utterance). S would only utter (23a) if H has done something to indicate that he is taking it as mutual that there is a Prince of Denmark, i.e. if S believes that H's discourse model contains an entity that can be referred to as the Prince of Denmark. For example, in line with Burton-Roberts' account, H might previously have said "*The Prince of Denmark smiled.*", or even "*The Prince of Denmark didn't smile.*": both of these show H is taking it as mutual that there is a Prince of Denmark. Similarly, S would only utter (23b) if H has done something to indicate that he is taking it as mutual that there is a Prince of Denmark and a play. For example, H might have just said "*The Prince of Denmark chose the play.*" or "*The Prince of Denmark didn't choose the play.*".

The consequence of this is that S will only utter, and so H will only find himself processing, (23a) if H has PRINCE-OF-DENMARK(Hamlet) in his discourse model. Similarly, if H finds himself processing (23b), H's discourse model will contain PRINCE-OF-DENMARK(Hamlet) and PLAY(P).

I will now look at the processing of (23a) in full, with the discourse model containing what I said it would contain, i.e., PRINCE-OF-DENMARK(Hamlet). The first clause, "The Prince of Denmark didn't smile", is processed in the way I have already described for presupposition-preserving negation to produce the logical form:

$$\texttt{PRINCE-OF-DENMARK(Hamlet)} \wedge \neg(\exists ev_2\ (\texttt{SMILE}(ev_2) \wedge \texttt{AGT}(ev_2, \texttt{Hamlet})))$$

Since I have assumed that PRINCE-OF-DENMARK(Hamlet) is in the hearer's discourse model, H is easily able to pick up the referent of the subject NP and then switch to negation mode for the rest of the utterance. This reading entails $\exists x$ PRINCE-OF-DENMARK(x).

The second clause of (23a), "there isn't a Prince of Denmark" is then processed. I handle "there is not" with hackery: I treat it as if it were a single word whose semantics are the same as "not"'s, i.e. it sets the NEG flag. With this flag set, a referent for the indefinite NP is not sought in the discourse model: Hamlet is not retrieved. Rather, a new entity for "a Prince of Denmark" is created, H2, and is labelled NEG, so it will be translated as an existentially quantified variable in the scope of negation. When the logical form for this clause is read off, we get:

$$\neg(\exists h_2\ (\texttt{PRINCE-OF-DENMARK}(h_2)))$$

But notice that the reading H gets for S's utterance is self-contradictory. The second conjunct contradicts the first conjunct. Note that this is T42's *initial* analysis. It has analysed both clauses of (23a) as object language utterances, and as such has found that to add them both to the discourse model would yield a contradiction: the set of possible worlds consistent with the discourse model would be empty. We will see below that this contradictory initial analysis is what leads T42 to consider a *re-analysis* in which the second clause is taken to be a metalinguistic utterance. On this re-analysis, there is no contradiction since the first clause is an object language statement and the second is a metalinguistic statement. Hence I am claiming that one possible clue for recognising metalinguistic utterances is that an initial object language analysis is contradictory.

Example (23b) similarly yields a contradiction on initial analysis. The discourse model contains PRINCE-OF-DENMARK(Hamlet) and PLAY(P) to begin with. The first clause, "*The Prince of Denmark didn't choose the play*", produces:

$$\texttt{PRINCE-OF-DENMARK(Hamlet)} \wedge \neg(\exists ev_3\ (\texttt{CHOOSE}(ev_3) \wedge \texttt{AGT}(ev_3,\ \texttt{Hamlet}) \wedge \texttt{PAT}(ev_3,\ \texttt{P}))) \wedge \texttt{PLAY(P)}$$

Then processing of the second clause, "*there wasn't a play.*", yields:

$$\neg(\exists p_2\ (\texttt{PLAY}(p_2)))$$

which gives a self-contradictory initial analysis.

This approach works for other presupposition triggers too, since these are also cases where entities must be found. For example, assume that H has (24a) in his discourse model, i.e. that there is a Lear and a fool and a hitting of the fool by Lear. Then S utters (24b). The first clause of (24b) would translate as (24c). The second clause would translate as (24d), giving a self-contradictory initial analysis.

(24) a. $\texttt{LEAR(L)} \wedge \texttt{FOOL(F)} \wedge \texttt{HIT(EV1)} \wedge \texttt{AGT(EV1, L)} \wedge \texttt{PAT(EV1, F)}$
b. "*Lear doesn't regret that he hit the fool because he didn't hit the fool.*"
c. $\texttt{LEAR(L)} \wedge \texttt{FOOL(F)} \wedge \neg(\exists ev_0\ (\texttt{REGRET}(ev_0) \wedge \texttt{AGT}(ev_0,\ \texttt{L}) \wedge \texttt{MOBJ}(ev_0,\ \texttt{EV1}))) \wedge \texttt{HIT(EV1)} \wedge \texttt{AGT(EV1, L)} \wedge \texttt{PAT(EV1, F)}$
d. $\neg(\exists ev_2\ (\texttt{HIT}(ev_2) \wedge \texttt{AGT}(ev_2,\ \texttt{L}) \wedge \texttt{PAT}(ev_2,\ \texttt{F})))$

That utterances such as those in (23) and (24) are initially found to be self-contradictory is perhaps the most important observation made by Burton-Roberts (see the previous section). This use of negation is marked or dispreferred in some way partly because it is so often signalled by contradiction. And it is this contradiction that triggers the re-analysis as a metalinguistic utterance.

Since these utterances can be used in these two ways (as object language and metalanguage statements), I will claim that they are *pragmatically ambiguous*. Stalnaker [1972] is the only person I know of who tries to elucidate the concept of a pragmatic ambiguity:

> "In general, a sentence has the potential for pragmatic ambiguity if some rule involved in the interpretation of that sentence may be applied either to the context or to the possible world. Applied to the context, the rule will either contribute to the determination of the proposition (as in the case of the referential use of definite descriptions) or it will contribute to the force with which the proposition is expressed. Applied to the possible world, the rule is incorporated into the proposition itself, contributing to the determination of a truth value." [p.394].

This is a useful way of looking at pragmatic ambiguity. It certainly captures the difference between referential and attributive uses of definite NPs (this is the other case of pragmatic ambiguity that we have seen in this thesis — see Section 5.2): referential uses are cases where a referent (discourse model entity) must be obtained in order to determine the proposition, attributive uses are cases where this is not so but the description is essential to determining the truth of the proposition in a possible world. This is not the distinction Horn is using when he talks of metalinguistic negation as a pragmatic ambiguity: narrow-scope negation determines the proposition for Horn, but his metalinguistic use of negation has nothing to do with truth against a possible world; instead it brings into question a previous utterance. So even if Horn's account of presupposition "cancellation" is "metalinguistic", he cannot relate it to Stalnaker's pragmatic ambiguity.

Recognising a contradiction on initial analysis is neither a necessary nor sufficient condition for recognition of a metalinguistic utterance. It is not a *sufficient* condition because detection of a contradiction need not always signal a metalinguistic utterance. For example, since a contradictory initial analysis flouts Grice's maxim of Quality, it might signal that a Gricean implicature should be generated. The circumstances under which we consider a metalinguistic reading rather than, e.g., a Gricean-style re-analysis are not yet clear.

And while a self-contradiction on the initial analysis might be a good clue to a metalinguistic utterance, it is not a *necessary* condition. For example, B's utterance in (25) uses metalinguistic negation but there is no initial self-contradiction:

(25) A: "*The Prince of Denmark smiled/didn't smile.*"
B: "*There isn't a Prince of Denmark.*"

I find B's utterance felicitous, if a little blunt.

Further, a contradiction is not a necessary condition because there are other, perhaps more direct, ways of signalling metalinguistic utterances. The most obvious of these is the use of the verb "exists". I am claiming that all of the following are metalinguistic for this reason, where (26c) is the case we might otherwise describe as presupposition "cancelling":

(26) a. "*The Prince of Denmark exists.*"
b. "*The Prince of Denmark doesn't exist.*"
c. "*The Prince of Denmark didn't smile; he/the Prince of Denmark doesn't exist.*"

In these cases we do not need an explanation in terms of an initial contradictory analysis being re-analysed: explicit use of "exists" signals a metalinguistic utterance directly.[16]

We must now look more closely at what a metalinguistic utterance is telling us. To facilitate this I will recap on the distinction which (following [Stalnaker 1972]) we made between the context (discourse model) and possible worlds.

The discourse model is simply a set of formulas of a logic and thus determines a set of possible worlds, those that are consistent with the formulas.[17] The proposition expressed by an utterance is fixed with reference to the discourse model. In particular, presuppositions must be satisfied by the discourse model. But once we have the proposition, we assess it for truth against a possible world, not against the discourse model. If we assess it against one of the worlds that are consistent with the discourse model, then the presuppositions will be true. But we need not choose one of these worlds; hence, presuppositions need not be true in the world we are using to evaluate the proposition. Once a presupposition has used the discourse model to fix the proposition, it has done its job. This is what allows an account of inaccurate reference, deception, etc.

Clearly, depending on the possible world chosen, the proposition representing the utterance reading may be true or may be false. If we allow partial models, where not everything is fully specified, as in [Fodor 1979], then we may also get cases where a proposition can be interpreted but the truth-value is 'undefined'. Whether we say such propositions are false, truth-valueless or have a third truth-value in that world becomes a matter of taste. This point is made by Russell in his reply to Strawson's paper on presuppositions: "[Strawson] considers that the word "false" has an unalterable meaning which it would be sinful to regard as adjustable, ... For my part, I find it more convenient to define the word "false" so that every significant sentence is either

[16] While T42 is set up to process the examples in (23) and (24), the present implementation does not handle the examples in (25) and (26). In (25) there is no contradiction for T42 to detect, and in (26) it is not clear that `EXISTS(Hamlet)` or `¬EXISTS(Hamlet)` would be appropriate representations for metalinguistic utterances.

[17] In this discussion, as elsewhere in the thesis, I should really bring in the notion of interpretation functions, relating symbols of the formulas to objects, relations and functions in a world. I have mostly ignored this in order to simplify the exposition.

true or false. This is a purely verbal question; and although I have no wish to claim the support of common usage, I do not think that [Strawson] can claim it either." [Russell 1957, pp.388-389]. My bias, like Russell's is towards a bivalent semantics but this really does not matter.

There may also be cases where, for the chosen world, a proposition cannot be interpreted. This would happen, for example, if the interpretation function does not define a denotation for a particular constant symbol appearing in the formula. Again it is a matter of taste whether we treat such cases as false, truth-valueless or having a third truth-value. I am going to claim that metalinguistic utterances are often concerned with cases such as this. They convey some indication of what the chosen world against which propositions are being evaluated for truth is like, or how it differs from some other world (often the 'real' world).

To make this more concrete, we will consider a conversation on the subject of Hamlet. We will assume the discourse model contains the proposition `PRINCE-OF-DENMARK(Hamlet)`. Propositions arising from a discourse are, I presume, evaluated for truth against some "distinguished" possible world, often the 'real' world. I will assume that propositions in our example discourse are evaluated against a partial possible world, which, in this case, *will* be consistent with the discourse model, hence `PRINCE-OF-DENMARK(Hamlet)` will be true in it,[18] and which also makes true `MAD(Hamlet)` and ¬`HAPPY(Hamlet)` but has nothing to say about whether `LIKES-CHEESE(Hamlet)` is true.

Clearly, if we try to answer the question "*Is the Prince of Denmark mad?*", we use the discourse model to obtain a referent for "the Prince of Denmark" and thus fix the proposition `MAD(Hamlet)`. In the world I described above, this is true, and so the answer to the question would be "*Yes*". "*Is the Prince of Denmark happy?*", i.e. `HAPPY(Hamlet)`, is false so the answer is "*No*", and "*Does he like cheese?*", i.e. `LIKES-CHEESE(Hamlet)`, is not specified and we might answer "*No*" or "*Don't know*" according to taste.

But suppose the respondent had been assuming a different distinguished possible world, in which there is no Prince of Denmark, and is asked "*Is the Prince of Denmark mad?*". Again the discourse model must contain `PRINCE-OF-DENMARK(Hamlet)`, otherwise the respondent would not find the referring expression acceptable at all. But on trying to evaluate `MAD(Hamlet)` in this world, there is no denotation for `Hamlet`, and so the reply might be "*The Prince of Denmark isn't mad because there is no Prince of Denmark*" or "*The Prince of Denmark isn't mad because the Prince of Denmark doesn't exist.*", i.e. a case of presupposition "cancellation".

The metalinguistic (presupposition "cancelling") utterance gives information about how the contents of the discourse model relate to the distinguished world which is being used to evaluate things for truth. Specifically, in these cases, the speaker of the presupposition "cancelling" utterance is

[18] It would have been possible to show inaccurate reference by having `KING-OF-FRANCE(Mitterand)` in the discourse model, hence utterances about the King of France produce propositions involving the constant `Mitterand`, but to have evaluated these in a possible world in which `KING-OF-FRANCE(Mitterand)` is not true but where `PRESIDENT-OF-FRANCE(Mitterand)` is true.

saying that the distinguished world she is assuming does not contain a denotation for `Hamlet`. So, I repeat, presupposition "cancelling" utterances (and, indeed, other metalinguistic utterances) say how the discourse model relates to the distinguished world against which the speaker is evaluating the truth of propositions.

This notion of metalinguistic utterances probably needs to be further worked out. But it seems to me that this kind of statement *is* qualitatively different from object language statements. For S to say that the Prince of Denmark is moody is simply to invite H to update his discourse model with `MOODY(Hamlet)`. Certainly this changes the possible worlds that are determined by the discourse model. But the distinguished world we are using to evaluate the truth of propositions need not be one of the worlds consistent with the discourse model (either before or after the update). Hence, it seems to me to be reasonable to posit metalinguistic utterances, which allow the speaker to indicate what her distinguished world is really like.

From the point of view of someone hearing such presupposition "cancelling" utterances, the metalinguistic nature of these would be detected in the way I have described earlier. The exact action to be taken is not entirely clear. If the hearer was assuming that the conversation was about the 'real' world, then if he accepts what the speaker has told him, he might have to change his conception of what the 'real' world is like. Specifically, if he originally thought there was a Prince of Denmark then he must now take the 'real' world to be a possible world in which there is no Prince of Denmark.

This does not stop S and H from talking about the Prince of Denmark: the discourse model entity `Hamlet` is still in H's discourse model and so can be referred to. And the truth of propositions about `Hamlet` will continue to depend on which world they are evaluated against.

Thus presupposition "cancelling" utterances describe denotations with respect to the distinguished world. They are instances of metalinguistic utterances, which should be more generally familiar. For example, if you are asked by a child whether Santa (really) exists then you must decide whether to continue the fiction that the 'real' world contains a Santa Claus ("*Yes, he exists.*") or otherwise ("*No, he doesn't really exist.*").

An important point about this discussion of the presupposition "cancellation" data is that it still insists that presuppositions be satisfied by discourse model entities. These are not cases of a presupposition being triggered and then withdrawn ("cancelled") or being allowed to fail. The presuppositions must be satisfied as usual.

Since it is not clear to me what action H must take on realising that S has said `Hamlet` (or the play, `P`, or the hitting of the fool, `EV1`) has no denotation in the distinguished world, I have not implemented anything along these lines. It would seem to require some way of saying who believes what is true in what world; representing this is something which I much earlier excluded as beyond the scope of the dissertation. But I am able to briefly discuss some of the possibilities.

The first thing I note is that S's utterance does not prevent further discourse from referring to the entity which has been said to be without a denotation in a particular world. This is not surprising since S and H may not be in agreement over which world propositions are being evaluated in. The following examples show this:

(27) A: "*The Prince of Denmark didn't smile; there isn't a Prince of Denmark.*"
B: *You're wrong. The Prince of Denmark does exist (and is smiling).*"

(28) A: "*The Prince of Denmark didn't choose the play; there wasn't a play.*"
B: "*If you ask Claudius, he will be able to confirm that the play took place last Thursday. So that shows how much you know about Danish court life!*"

(29) A: "*Lear doesn't regret he hit the fool; he didn't hit the fool.*"
B: "*Lear did hit the fool: it happened last Thursday (after the play).*"

The underlined expressions can all pick up the entity which S said has no denotation in the distinguished world.[19]

Equally even S can continue to refer to the entity despite having said it has no denotation in the distinguished world:

(30) A: "*The Prince of Denmark didn't smile; there isn't a Prince of Denmark.*"
B: *You're wrong. The Prince of Denmark does exist (and is smiling).*"
A: "*Well ... if he exists, he smiled, but I still think he's a figment of your imagination.*"

This suggests that whatever else H does after a metalinguistic utterance, he should not remove the entity from the discourse model. Rather, what seems to be necessary is to have representations of different possible worlds in the knowledge base and to use metalinguistic utterances to, e.g., make you change your mind about which one represents the 'real' world, and possibly to make you evaluate future discourse, e.g. answer questions, against a different possible world representation than the one you were using up to now. Beyond this I am not sure what should happen to the entity: I have not looked into the representation issues at this level of detail. Something must show what S believes the various worlds to be like. And according to S's credibility or otherwise, we might also need to show whether H agrees with S's judgement. This information is quite sophisticated as lack of denotation in one distinguished world does not imply lack of denotation in another world:

[19] In all of the examples the underlined expressions are "given" environments and so demand the entity be retrieved, except in (29) where the first underlined expression is not in a "given" environment but is still anaphoric in the way I have shown verbs can be — see Section 6.3.

(31) A: "*The Prince of Denmark isn't mad; there isn't a Prince of Denmark.*"
B: *No, of course there isn't really, but you agree he's mad in the play?*"
A: "*Oh yes, he's off his head in the play.*"

There remains some data that I have not yet accounted for: the asymmetry between "cancellation" in negative and positive sentences. It is alleged that "cancellation", i.e. a metalinguistic utterance, cannot felicitously follow a positive sentence:

(32) *"*The Prince of Denmark smiled but there isn't a Prince of Denmark.*"

How am I to account for this? First I will cast at least some doubt on the data. I find the following perfectly acceptable:

(33) "*The Prince of Denmark smiled, but there isn't really a Prince of Denmark.*"

It seems that what is happening here is a switch of distinguished worlds. We are supposed to evaluate the first conjunct of (33) against a possible world such as one that reflects the characters of Shakespearean plays, and in this case, it is, in these plays, true or false as the case might be. The second conjunct tells us that the entity has no denotation in the real world (hence the use of "really"). The reason these so often sound infelicitous (as in (32)) is that these switches are rare and need to be made clearer by using words (such as "really") that show that the possible world being used has changed.

To summarise this section, in T42 negation is semantically unambiguous: it is internal (narrow-scope), so presuppositions are always triggered and must be satisfied. Traditional cases of presupposition "cancellation" often involve the speaker making a self-contradictory utterance to indicate a metalinguistic utterance. But there are other ways of signalling such utterances, e.g. by using the verb "exists". But, even in these cases of presupposition "cancellation", the presuppositions must still *always* be satisfied. That is to say that the *hearer* must be able to find a referent in his discourse model (even though the *speaker* is going on to say that she does not think that the entity has a denotation in a distinguished world). These are *not* cases of presupposition failure. If the presupposition fails, i.e. the necessary entities are not in H's discourse model, then that reading of S's utterance is unacceptable, just as it would be in a positive utterance. Presupposition failure does not license the presupposition-cancelling reading of the negative utterance.

A question that was raised in the previous chapter was: how long will something remain felicitously "cancellable"? To some extent this question was shown to be a bogus one that arises when presuppositions are not seen as preconditions. However, in so far as it is a real issue, the answer for T42 in principle is that something will be "cancellable" forever. S and H can talk about an entity for as long as they like and then either of them might, in principle, indicate that the entity does not have a denotation in some distinguished possible world. Often presupposition "cancellation" would be used to signal a shift from discussion of one possible world (e.g. where the events in a play are true) to another (e.g. the real world).

::::

This chapter has reviewed five different accounts of the interaction between presuppositions and negation. All but the last (Burton-Roberts') was shown to be unsuitable for use within T42 for either theoretical or technical reasons. Burton-Roberts' account took negation to be semantically unambiguous and to always preserve presuppositions. T42 takes the same line: presuppositions are always triggered and must be satisfied by the discourse model. Both accounts also offer a pragmatic explanation of presupposition "cancellation". Burton-Roberts argues that these cases are cases of metalinguistic negation, where the speaker objects to a previous utterance on the basis of its form. I have argued that there is a division in his data between those cases of metalinguistic negation in which the previous utterance's form is objected to and those where a proposition is objected to. This has led me into giving a different pragmatic account in T42: these pragmatic uses are metalinguistic in the different sense that they say that a discourse model entity has no denotation in a distinguished world. This has several advantages given in the text; it is much more in line with Stalnaker's conception of pragmatic ambiguity, it brings these cases into line with other such uses (e.g. "*The Prince of Denmark exists.*") and explains why "cancellation" in positive sentences might just sometimes be possible. In the next chapter, I move on to looking at the presuppositions of utterances of complex sentences. My account of the data there lends further credence to my account here since it will again be demanded that presuppositions are always satisfied by the discourse model.

Chapter 9

T42 and the Presupposition Projection Problem

In this chapter I present the *projection problem* for presuppositions and its solution in T42. Section 9.1 briefly presents the projection problem as it has traditionally been defined: the utterance of a *compound sentence* (i.e. a sentence that itself has sentential constituents) might not inherit all the presuppositions triggered by the sentences embedded within it; some of the embedded presuppositions seem to get "cancelled" or "suspended". Section 9.2 gives an overview of the solution T42 exploits to handle the projection problem. I suggest that embedded presuppositions are not "cancelled" or "suspended". They are still triggered and must still be satisfied, and if they are not satisfied that utterance reading does not come off. However, these presuppositions may be satisfied by things internal to the compound sentence (i.e. in the immediate linguistic context); in this sense, they do not 'reach the surface' as presuppositions of the utterance as a whole, i.e. they do not demand satisfaction in the non-immediate context. Section 9.3 gives a comprehensive review of projection problem data and shows how naturally T42 can make correct predictions about this data. However, there are some residual cases which, while they seem to require a treatment compatible with T42's approach, cannot yet be handled in T42.

Reviewing other 'solutions' to the projection problem is left to Chapter 10. While it would perhaps be more conventional to review other approaches *before* describing my own work, the unconventional ordering used here allows me to give a comprehensive review of projection data and T42's approach to this data side by side in this chapter. Then in Chapter 10 I can select relevant examples only from the Chapter 9 data to illustrate specific weaknesses of other accounts, and to highlight the differences between T42 and these other accounts.

9.1 The Projection Problem Defined

The projection problem was originally defined by Langendoen & Savin as "...the question of how the presuppositions and assertions of a complex sentence are related to the presuppositions and assertions of the clauses it contains." [Langendoen & Savin 1971, p.55]. Langendoen & Savin were responsible in the same paper for the much quoted 'wrong answer' to this problem: they said that a complex sentence would inherit all the presuppositions of its constituent clauses. There is, as the rest of this chapter shows, a lot of data where this does not, in any simple sense, hold true. However, my view is that the problem is a somewhat bogus: it only arises in the way that it does when presuppositions are viewed as ordinary inferences. My view of presuppositions as preconditional inferences, and T42's incremental processing of utterances (working sequentially through a sentence from left-to-right), overcome many of the difficulties of projection.

When presuppositions are not viewed as preconditions, a projection algorithm is needed to "cancel" or "suspend" certain presuppositions of embedded sentences (those which other parts of the utterance or context indicate are not being taken for granted); the algorithm must also make sure that in cases where a proposition is both asserted or entailed by one part of an utterance and presupposed by another, the proposition does not get labelled as a presupposition. As I will show, both of these requirements of a projection algorithm vanish if a presupposition is just a precondition that must be satisfied by the discourse model in order to get an utterance reading.

There are two main classes of construct which embed sentences within others sentences: subordinating constructions (e.g. verbs which take sentential complements, cleft constructions and sentential adverbs) and coördinating constructions (conditionals, conjunctions and disjunctions). I will consider the problems each presents separately.

9.1.1 Subordinating Constructions

Presupposition triggers may occur in the sentential complements of verbs of saying and verbs of propositional attitude[1]. Ordinarily (1a) presupposes (1b), i.e. for a hearer to find (1a) acceptable, he must be able to retrieve from his discourse model an entity which is a dog:

(1) a. "*Proteus hit the dog.*"
b. 'There is a dog.'

Does the same apply to (2a), (2b) and (2c), in which (1a) is embedded under a selection of verbs that subcategorise for sentential complements?

(2) a. "*Launce regrets that Proteus hit the dog.*"
b. "*Launce believes that Proteus hit the dog.*"
c. "*Launce said that Proteus hit the dog.*"

I would say that it does in all three cases: a dog must be found or inferred in each. And in each case, the name "Proteus" in this complement also

[1] I will ignore clefts and sentential adverbs in the presentation, but what I have to say applies equally well to them.

has a presupposition that a discourse model entity called "Proteus" can be identified. (There is an additional presupposition in (2a) due to the factive, "Launce regrets", that the discourse model contains an event involving the dog being hit by Proteus).

What is true of definite NPs embedded under such verbs also seems to be true of other presupposition triggers embedded under such verbs. For example, (3a) presupposes all of (3b), (3c), (3d) and (3e). These presuppositions are preserved when (3a) itself is embedded into another sentence as in (4a), (4b) and (4c):

(3) a. "*Launce regretted that Proteus hit the dog.*"
b. 'There is an entity called Launce.'
c. 'There is an entity called Proteus.
d. 'There is an entity which is a dog.'
e. 'There is an entity which is the hitting of the dog by Proteus.'

(4) a. "*Valentine regrets that Launce regretted that Proteus hit the dog.*"
b. "*Valentine believes that Launce regretted that Proteus hit the dog.*"
c. "*Valentine said that Launce regretted that Proteus hit the dog.*"

(Due to the factive, "Valentine regrets", (4a) also presupposes that the discourse model contains an entity confirming that Launce regretted the hitting event described by (3e)).

There has been a lot of debate about this data. In Chapter 10, I mention the work of Karttunen & Peters [1979], who do not agree that (1b) is a presupposition of (2b) and (2c), or that (3b), (3c), (3d) and (3e) are presuppositions of (4b) and (4c). On the other hand, it would seem that most people now follow Gazdar [1979], who claims that the presuppositions survive in all these cases (unless there are other contextual factors that dictate otherwise). I believe that Gazdar is right, i.e. the presuppositions survive, but that there are two specific exceptions to this, involving verbs of propositional attitude such as "believe" and involving verbs of saying such as "say". I will look at verbs such as "believe" first.

The only cases where presuppositions embedded under verbs such as "believe" might not survive are cases of presuppositions of definite NPs. All other presupposition triggers will still have to be satisfied. For example, just as (2b) presupposes (1b), (5a) presupposes (5b), (5c) presupposes (5d) and (5e) presupposes (5f). It is only (5g) that need not presuppose (5h):

(5) a. "*Launce believes that Caliban managed to find a scrap to eat.*"
b. 'Caliban tried to find a scrap to eat.'
c. "*Launce believes that Lear stopped beating his fool.*"
d. 'Lear had been beating his fool.'
e. "*Launce believes that Lear hit the fool again.*"
f. 'Lear had hit the fool before.'
g. "*Launce believes the best doctor could save that patient.*"
h. 'There is a best doctor.'[2]

[2] I.e. there is an identifiable individual represented by a discourse model entity who is the best doctor.

The presuppositions of embedded implicative verbs (5a), aspectuals (5c) and iteratives (5e) do survive, but those of definite NPs (5g) need not.

But this should not be surprising; it is something that we have looked at already. It is, in fact, nothing to do with whether the presupposition trigger is embedded within the complement of a verb of propositional attitude. Rather, it is straightforwardly another manifestation of the fact that definite NPs may be used referentially or attributively. When they are used referentially, they have presuppositions: we must identify a discourse model entity as their referent. When they are used attributively, this need not be so. I believe that the only case where a presupposition trigger embedded under a verb such as "believe" can lose its presupposition is the case of a definite NP being used attributively, and that the reason that we might be misled into believing this to be a property of the verb of propositional attitude is, as Cole [1978] suggests, that the complement of a verb such as "believe" is one syntactic environment which 'encourages' attributive uses.[3] Thus in (5g), if the NP "the best doctor" is being used referentially there is a presupposition; if it is being used attributively, there is not.[4] Of course, this still leaves the very difficult, and unresolved, problem of distinguishing an attributive use from a referential use (see Section 5.2).

There are also cases where presuppositions embedded under verbs of saying such as "say", "tell", "mutter", "order", "whisper" and even "criticise" and "accuse" may not retain their presuppositions. This possibility applies to all presuppositions, not just those associated with definite NPs. For example, it is easy to imagine contexts in which (2c) and (4c) do not retain the presuppositions I attributed to them earlier. Again it is [Cole 1978] that offers an explanation. Cole says that these verbs are lexically ambiguous: they may be used both to paraphrase what someone else has said, and to *quote* someone else's utterance. The quotation may be direct (6a) or indirect (6b). In both quotational cases it is clear that the speaker is reporting the actual words (or nearly so) used by another person with no commitment to them herself. Hence, the hearer need not expect the presuppositions of the quoted sentence to be satisfied.[5]

(6) a. "*Launce said "Proteus hit the dog.""*
b. "*Launce said (that) Proteus hit the dog.*"

A problem that I have not looked at, but that will need to be solved if we are to make progress on these verbs of saying is, of course, that indirect

[3] Cole similarly explains, i.e. with reference to the referential/attributive distinction, why these environments can be referentially opaque, i.e. why equivalent descriptions sometimes cannot be substituted in these environments. If he is right, referential opacity no longer needs an account in terms of a scope ambiguity. Substitution of equivalent descriptions will not be possible precisely when a description is used attributively, because in such uses the description is essential to what the speaker is saying, and is not being used simply to identify some referent.

[4] Note that in (5g) there is another definite NP, "that patient". The presupposition that there is a patient survives because the definite NP, by virtue of being a demonstrative, is almost certainly being used referentially.

[5] While I believe Cole's account is along the right lines, it is a little unattractive to have to posit a lexical ambiguity to account for these cases.

quotational uses are indistinguishable in their syntactic form from other uses of verbs of saying: how will a computational system detect one use rather than the other?[6]

9.1.2 Coördinating Constructions

In producing a reading for the utterance (7a), the presupposition (7b) will be triggered:

(7) a. "*Launce's dog likes him.*"
    b. 'There is a dog owned by Launce.'

i.e. in T42, there will be a precondition on the hearer understanding a reading of (7a) that such a dog be in the discourse model. (Of course, using "owned" here is a bit strong. I am merely choosing one possibility and not concerning myself too much with the exact form needed). Utterance of this simple sentence will be acceptable if an entity, `Crab` say, is found. Otherwise the reading will not come off. The reading of (7a):

$$\mathtt{LAUNCE(L)} \land \mathtt{DOG(Crab)} \land \mathtt{OWNS(EV0)} \land \mathtt{AGT(EV0,\ L)} \land \mathtt{PAT(EV0,\ Crab)} \land \mathtt{LIKES(EV1)} \land \mathtt{AGT(EV1,\ Crab)} \land \mathtt{PAT(EV1,\ L)}\ .$$

entails

$$\exists x \exists y\ (\mathtt{DOG}(x) \land \mathtt{OWNS}(y) \land \mathtt{AGT}(y,\ \mathtt{L}) \land \mathtt{PAT}(y, x)).$$

Now we will look at what happens if (7a) is embedded into compound sentences. In the interests of giving more natural sounding examples, I will freely use "*his dog*" in place of "*Launce's dog*" as appropriate.

(8) a. "*Launce owns a dog and his dog likes him.*"
    b. "*If Launce owns a dog, his dog likes him.*"

These are simple examples, introduced here merely to facilitate this discussion. A comprehensive set of examples will be reviewed later. It should be clear that (8a) still entails $\exists x \exists y\ (\mathtt{DOG}(x) \land \mathtt{OWNS}(y) \land \mathtt{AGT}(y,\ \mathtt{L}) \land \mathtt{PAT}(y, x))$, but that (8b) does not entail this. In accounts in which presuppositions are viewed as ordinary inferences, the presupposition triggered by the consequent in (8b) must be "cancelled" or "suspended". But also in these accounts, for (8a) it is necessary to make sure that this proposition is not labelled as both an assertion (from the first conjunct) *and* a presupposition (from the second conjunct). In capturing these facts in T42, I do not want to introduce special mechanisms for inhibiting presupposition triggers, "suspending" presuppositions or "relabelling" presuppositions. Further, to keep my treatment of presuppositions entirely uniform for simple positive, negative *and* complex utterances, every presupposition trigger *must* generate its

[6] One clue, of course, might be presupposition failure itself, which will raise the possibility of a quotational use.

presupposition and every presupposition that is generated *must* be satisfied by finding an appropriate entity in the discourse model.

However, in the examples we have seen in previous chapters, it so happens that if the presuppositions of an utterance reading were satisfied, then, for worlds consistent with the discourse model, the presupposition was also an entailment of the utterance reading. To handle the data in (8b), however, I will need to ensure that the reading produced does not necessarily entail the presupposed proposition even as far as worlds consistent with the discourse model are concerned. It will be in this sense that a presupposition does not get projected (i.e. it gets "blocked", "inhibited", "retracted", "neutralised", "cancelled", "suspended" or whatever). Nonetheless, *there will still be a precondition that must be satisfied by the discourse model.*

If I can give an account in these terms this will again show the advantages of treating presuppositions as preconditions. If presuppositions were instead to be treated as ordinary inferences, then some sort of recursive projection algorithm would be needed to decide whether or not the inference should be "cancelled" and to work out how to label a proposition that is both entailed and presupposed. I explain my account in the next section.

9.2 Overview of T42 and Projection

Before I describe yet further changes to T42, I will recap on its basic processing as detailed in earlier parts of the thesis. The parser sends constraints to the constraint satisfier, constituent by constituent, as it works from left-to-right through the input. Each constraint adds to a Constraint Satisfaction Problem (CSP) which the constraint satisfier tries to solve. If the constraint satisfier cannot solve the problem, it informs the parser that an alternative analysis should be tried.

To solve the CSP, the constraint satisfier needs discourse model entities. Where the constraint satisfier gets these from will depend on what kind of 'environment' (part of the utterance) the constraint was generated by. In "given" environments (where the `GIVEN` flag is set) entities must come from the discourse model: in "new" environments (where the `GIVEN` flag is not set), entities may come from the discourse model and are also newly created. In the scope of negation (where the `NEG` flag is set), if the `GIVEN` flag is also set, then entities must come from the discourse model, but if the `GIVEN` flag is not also set, then entities must be created: these entities will not be accessible to referring expressions since they are existentially quantified variables in the scope of negation.

The discourse model is split into two: the immediate linguistic context (ILC) and the non-immediate context (NIC). The ILC is a 'scratchpad' for the constraint satisfier and represents its workings on the current utterance. When looking for an existing entity, the constraint satisfier may use entities from either the ILC or the NIC. So far in this thesis the division between the ILC and the NIC has been mostly irrelevant. In this chapter, however, this division is given an important role to play.

9.2.1 The General Approach

As explained above, while the GIVEN flag is set (i.e. while processing a presupposition trigger), T42 will try to find existing entities. This behaviour will not be inhibited in complex sentences. When processing these sentences, T42 behaves as usual but, in certain cases, will be able to pick up 'special' entities. (I will say what these are in Subsection 9.2.2). That is to say, in cases where a presupposition should, under traditional accounts, be cancelled, T42 will manage to pick up an entity as usual, but the logical inference rule of existential generalisation will not be possible from the logical form built using this entity. Thus the final logical form will not entail the presupposition. In this way we have the effect of presupposition "cancellation" in complex sentences, without any special "cancellation" mechanisms.

For example, in (8a) ("*Launce owns a dog and his dog likes him.*"), the first conjunct sets up a discourse model entity for the dog which is then picked up by the presupposition that there be a dog which is triggered in the second conjunct. There is no relabelling needed: the presupposition was triggered as normal and satisfied as normal (except that it was satisfied by something from within the same utterance). Similarly in (8b) ("*If Launce owns a dog, his dog likes him.*"), the antecedent can set up an entity which can be picked up by the consequent's presupposition. Thus in both cases the presupposition has found an entity and so has been satisfied. However, the entity created by the first conjunct in (8a) is a *bona fide* entity, whereas in (8b) it is only a *hypothetical* entity (see Subsection 9.2.2). What I mean by this is that (8a)'s entity is a normal constant and so the fact that Launce owns a dog will be entailed, but (8b)'s entity is a quantified variable introduced in the antecedent of a conditional: existential generalisation will not apply to the logical form and so there will be no spurious entailment.

It can be seen from this description that an approach that exploits the ordering of clauses in a discourse will be important because the felicity of a clause may depend on whether its presuppositions can be satisfied by entities introduced in earlier clauses. An incremental system would seem to be a suitable way of achieving these effects.

These remarks are suggestive of Kamp's Discourse Representation Theory (DRT) [Kamp 1984] as described in Section 6.6 (and Van der Sandt makes the same analogy towards the end of [Van der Sandt 1987]). Since DRT has some of the 'flavour' of T42's projection problem solution and also the advantage of a diagrammatic notation, I will pursue this analogy further in this section.

In DRT we could represent (8a) as (9):

(9)

$$\begin{array}{|lll|}\hline
p\ q\ r\ s\ t\ u\ v\ w & & \\
 & & \\
\mathtt{Launce}(p) & \mathtt{Owns}(q) & \mathtt{Agt}(q,p) \\
\mathtt{Pat}(q,r) & \mathtt{Dog}(r) & \\
 & & \\
\mathtt{Male}(s) & \mathtt{Owns}(t) & \mathtt{Agt}(t,s) \\
\mathtt{Pat}(t,u) & \mathtt{Dog}(u) & s=p \\
t=q & u=r & \mathtt{Likes}(v) \\
\mathtt{Agt}(v,u) & \mathtt{Pat}(v,w) & \mathtt{Male}(w) \\
w=s & & \\ \hline
\end{array}$$

Here the two clauses of the conjunction have augmented the same DRS, and anaphoric relations have been set up from the reference marker s to p, etc. These anaphoric relations are permissible because the markers are at the same level in the structure. The definition of truth for this DRS is straightforward (as described in Section 6.6), i.e. if there is a function that can map this structure into a model (in the model-theoretic semantics sense), observing all the constraints, then the DRS is true.

Representing (8b) is more complicated. We get (10):

(10)

$$\text{A}\ \begin{array}{|l|}\hline
p \\
 \\
\mathtt{Launce}(p) \\
 \\
\text{B}\ \begin{array}{|l|}\hline q\ r \\ \\ \mathtt{Owns}(q) \\ \mathtt{Agt}(q,p) \\ \mathtt{Pat}(q,r) \\ \mathtt{Dog}(r) \\ \hline\end{array} \quad \Rightarrow \quad \text{C}\ \begin{array}{|lll|}\hline s\ t\ u\ v\ w & & \\ & & \\ \mathtt{Male}(s) & \mathtt{Owns}(t) & \mathtt{Agt}(t,s) \\ \mathtt{Pat}(t,u) & \mathtt{Dog}(u) & s=p \\ t=q & u=r & \mathtt{Likes}(v) \\ \mathtt{Agt}(v,u) & \mathtt{Pat}(v,w) & \mathtt{Male}(w) \\ w=s & & \\ \hline\end{array} \\
 \\ \hline
\end{array}$$

With complex DRSs there is a partial ordering on the nested DRSs. This ordering defines the accessibility of markers for anaphoric relations. The partial ordering on the DRSs in (10) is $A > B > C$. Anything accessible to A is accessible to B and anything accessible to B is accessible to C. Hence C was able to equate s with p and so on.

The truth definition for (10) is that (10) will be true if and only if *every* function f that finds in the model a dog owned by Launce can be extended to a function f' that also finds a dog owned by Launce which likes Launce, and the two dogs owned by Launce that are found must be the same. In other words, for all of Launce's dogs that you can find, then they all like him. Crucially, the presupposition trigger in the consequent of (8b) has been satisfied, i.e. "his dog" has found an entity to satisfy it, this being the dog introduced in the antecedent of the conditional ($u = r$), but, given the way that truth is defined, the whole utterance reading does not demand the existence of a *bona fide* dog in order to be true.

The DRT accessibility constraints say that markers introduced in subsequent utterances are not able to be anaphorically related to markers from the inner boxes. So while we can extend (8a) to (11):

(11) *"Launce owns a dog and his dog likes him. It barks a lot."*

by putting more constraints into the box in (9) and making the marker for "it" equal to u (or r), we cannot felicitously extend (8b) to (12):

(12) *"*If Launce owns a dog, his dog likes him. It barks a lot.*"[7]

If we draw up the DRS we get (13):

$$\text{A}\ \boxed{\begin{array}{l} p \\ \\ \texttt{Launce}(p) \\ \\ \text{B}\ \boxed{\begin{array}{l} q\ r \\ \\ \texttt{Owns}(q) \\ \texttt{Agt}(q,p) \\ \texttt{Pat}(q,r) \\ \texttt{Dog}(r) \end{array}} \ \Rightarrow\ \text{C}\ \boxed{\begin{array}{lll} s\ t\ u\ v\ w & & \\ & & \\ \texttt{Male}(s) & \texttt{Owns}(t) & \texttt{Agt}(t,s) \\ \texttt{Pat}(t,u) & \texttt{Dog}(u) & s=p \\ t=q & u=r & \texttt{Likes}(v) \\ \texttt{Agt}(v,u) & \texttt{Pat}(v,w) & \texttt{Male}(w) \\ w=s & & \end{array}} \\ \\ \texttt{Barks-a-lot}(x) \\ {}^{*}x=u \end{array}} \tag{13}$$

Marker x cannot pick up u (or r). The end of the conditional closes off the right-hand side inner box (C), so subsequent sentences are not able to pick up reference markers from within the boxes.

It is now well-recognised that this is a simplification. Counter-examples are discussed under the heading of "modal subordination" [Roberts 1987]. An example would be:

(14) *"If Launce owns a dog, his dog likes him. It would bark a lot."*

The use of a modal verb in the second utterance indicates that the speaker is still talking about how things might be if the antecedent of the conditional were to be true. In DRT terms, her utterance extends box C in (13), and so she may still refer to the dog (u or r). The circumstances in which this "modal subordination" may take place are not clearly defined. For this reason, further consideration of such cases is beyond the scope of this thesis. I will continue to assume that the end of a conditional closes the inner box making the markers within the boxes inaccessible.

This description of DRT gives some of the flavour of the solution I have adopted in T42. Thus, I describe below how I have extended T42 to handle conditionals and conjunctions in a way that is similar to that used in DRT.

[7] Actually I will later be suggesting that conditionals are pragmatically ambiguous between the categorial assertion of a condition, on which reading (12) is anomalous, and another reading for which (12) might not be anomalous.

9.2.2 Some of the Details

In what follows I will be explaining the difference between what I have described as "hypothetical discourse model entities" and "*bona fide* discourse model entities". These labels were intended as an expository device. Unfortunately they have led to some confusion. To preclude this confusion from arising, I wish to stress that the discourse model (comprising both the ILC and the NIC) contains nothing more nor less than a set of formulas of logic. It so happens that the information in the ILC is stored in a data structure more amenable to use by the constraint satisfier and must be converted to a more usual form when, at the end of processing an utterance, a reading is moved from the ILC to the NIC. But this conversion does not affect the model-theoretic interpretation of any of the information.

The notions of hypothetical and *bona fide* discourse model entities are used when explaining the term "accessibility". In DRT and T42, "accessibility" describes which reference markers are candidate antecedents for referring expressions. In T42 candidate antecedents are the discourse model entities. In particular, any constant symbol (denoting an object in a model), whether it be in the ILC or the NIC is an accessible antecedent.[8] These constant symbols in the discourse model, I refer to as "*bona fide* discourse model entities".

However, to handle cases such as (8b) (whose DRS is given in (10)), we have to allow referring expressions in, e.g., the consequent of a conditional to co-refer with universally quantified expressions in the antecedent of the conditional. In other words, such a referring expression is translated as a mention of a universally quantified variable. To emphasise that these variables are accessible referents and, as such, may appear in satisfaction sets in the CSP, I refer to them as discourse model entities. But to emphasise that they do not denote objects in a model (since they are universally quantified variables), I prefer to refer to them more specifically as "hypothetical discourse model entities".

Note though that this does not apply to universally quantified variables in the NIC. These are not accessible; they cannot be used as anaphoric antecedents; they are not discourse model entities (neither *bona fide* nor hypothetical).

Hence you can see that to implement T42's analogue of DRT's accessibility constraints, I have made use of the difference between the ILC and the NIC. While T42 processes the current utterance, if it needs an entity it may use entities that are in the ILC or in the NIC. When processing finishes and a reading for the utterance has been found, the reading is transferred out of the ILC into the NIC. In making this transfer, the data structures change from being a list of constraints on entities to being a formula of logic. It is in converting these constraints on entities into a logical form that *some* of the entities will effectively be made inaccessible. Since newly created entities within a conditional, for example, may be universally quantified variables,

[8] If T42 distinguished different levels of salience or used a notion of focus, then this might need to be more restrictive.

something that was an *entity* in the ILC and so could be used to satisfy presuppositions, may become inaccessible in the NIC and so will not be retrieved in future searches for discourse model entities. But its model-theoretic interpretation as a universally quantified variable does not change.

Accessibility is thus left in the hands of the user of T42, i.e. the speaker, S. If the speaker conjoins (using "and") further utterances to the current utterance, the ILC will not be emptied, so hypothetical entities (universally quantified variables) will remain as entities for the time-being, and will therefore be accessible. When the speaker starts a new utterance on the other hand, the ILC will be emptied into the NIC, and hypothetical entities (variables in the NIC) will no longer be accessible.

I will go over examples (11), (12) and (14) to indicate how all this works. In example (11) ("*Launce owns a dog and his dog likes him. It barks a lot.*"), the dog entity created in the first conjunct can be picked up by "his dog" in the second conjunct. On reading the full stop, the ILC is emptied; the logical form put into the NIC will contain a *bona fide* entity for the dog which can then be accessed as the referent for "it" in the second utterance. However, in example (12) (*"*If Launce owns a dog, his dog likes him. It barks a lot.*"), the dog created in the antecedent is a (hypothetical) entity while it is in the ILC and so can satisfy the search for a dog in the consequent, but when the contents of the ILC are transferred to the rest of the discourse model on finishing the utterance, we see more clearly that the dog entity is in fact a variable, and this cannot be found as the referent for "it" in the second utterance.

Unfortunately, example (14) ("*If Launce owns a dog, his dog likes him. It would bark a lot.*") will be handled in the same way as example (12). Contrary to intuitions, therefore, "it" in the second utterance cannot pick up the dog introduced in the first utterance. If the user wants to signal that she would like "it" to be able to access the dog entity, she must type in (15) instead of (14):

(15) "*If Launce owns a dog, his dog likes him and it would bark a lot.*"

By continuing the first utterance using a conjunction, the dog entity remains in the ILC and so can be picked up by the "it". I believe that, given that the conditions under which modal subordination, as in (14), is possible are not clear, the idea of leaving accessibility under user control in this way would seem to be a reasonable thing to do at present.

With accessibility described, I will now look at subordinating constructions, and at each of the coördinating constructions, and discuss the logical forms they will produce. This is prior to looking at coördinating constructions again alongside a comprehensive set of examples in the next section.

- **Subordinating Constructions**

Presuppositions embedded under subordinating constructions will be triggered as normal and must be satisfied by entities already in the discourse model (these entities may of course have been set up by the previous discourse or by the physical discourse setting, or may have been added to the

ILC by earlier parts of the utterance). T42 cannot handle the cases where these presuppositions are supposedly lost, as noted in Section 9.1.1, i.e. verbs of saying used in quotational ways and attributive uses of definite NPs in opaque (and other) environments. The latter are outside the scope of T42 because they are attributively used NPs, which I have had to exclude for the reasons given in Section 5.2. The former are excluded because I am not sure what logical form quotational uses should receive. Whatever it is, they would need to inhibit presupposition triggers (i.e. inhibit the $\star$`GIVEN` constraint).

Dealing with subordinating constructions in general also raises problems, independent of those relating specifically to presuppositions, concerning what logical form to give to utterances containing these constructions. I have not looked into these problems in any detail and have just gone for a very simple logical form. Here are the lexical entries for "regret" (which was given before in Chapter 7), "believe" and "say":

word	*category*	*meaning*
"regret"	$(Sx\backslash NPy)/Sz$	REGRET(x) AGT(x,y) $\star$GIVEN(z) MOBJ(x,z)

"believe"	$(Sx\backslash NPy)/Sz$	BELIEVE(x) AGT(x,y)
		MOBJ(x,z) OPAQUE(z)

"say"	$(Sx\backslash NPy)/Sz$	SAY(x) AGT(x,y)
		MOBJ(x,z) OPAQUE(z)

I will not discuss "regret" again (see Section 7.3.2). "Say" and "believe" do not differ from one another from a presuppositional point of view (given that I am ignoring the cases I mentioned earlier where they lose presuppositions), so I will concentrate on "believe" alone.

If T42 processes an utterance such as "*Launce believes that Proteus hit the dog.*" in a context containing LAUNCE(L), PROTEUS(P) and DOG(D), then it will create a belief event, EVO, say, and a hitting event, EV1. The logical form will be:

$$\texttt{LAUNCE(L)} \wedge \texttt{BELIEVE(EVO)} \wedge \texttt{AGT(EVO, L)} \wedge \texttt{MOBJ(EVO, EV1)} \\ \wedge \texttt{OPAQUE(EV1)} \wedge \texttt{PROTEUS(P)} \wedge \texttt{HIT(EV1)} \wedge \texttt{AGT(EV1, P)} \wedge \\ \texttt{PAT(EV1, D)} \wedge \texttt{DOG(D)}$$

Once this is added to the discourse model, the hitting event should be inaccessible. To achieve this effect, the above ought to be translated into something more like the following:

$$\texttt{LAUNCE(L)} \wedge \texttt{Bel}_L(\exists ev_1\ (\texttt{HIT}(ev_1) \wedge \texttt{AGT}(ev_1\texttt{, P}) \wedge \texttt{PAT}(ev_1\texttt{,} \\ \texttt{D))))} \wedge \texttt{PROTEUS(P)} \wedge \texttt{DOG(D)}$$

Since the inference engine cannot handle this higher-order logical form, I insert the first form above instead, but to make entities such as EV1 inaccessible, I do not treat entities labelled OPAQUE as discourse model entities.[9]

[9] This is an inelegant hack.

- **Conjunctions**

The processing of conjunctions in T42 should require little explanation by now. The clauses are processed in order of occurrence; presuppositions are triggered and discourse model entities are searched for. As both the ILC and the NIC can be consulted in the search for an entity, the utterance reading will come off if an entity is found, otherwise it will not come off. Thus, the first clause in a conjunction can set up entities which can be used to satisfy the presuppositions of the second clause (as in (8a)). The lexical entry for "and" is:

word	*category*	*meaning*
"and"	$(S_x/S_y)/S_z$	-

For an utterance such as "*Launce owns a dog and his dog likes him.*" in a context containing LAUNCE(L), we get constraints as follows:

```
LAUNCE(L) ∧ OWNS(EV0) ∧ AGT(EV0, L) ∧ PAT(EV0, D) ∧ DOG(D)
∧ LIKES(EV1) ∧ AGT(EV1, D) ∧ PAT(EV1, L)
```

Unless the conjunction is embedded into some other construction, all these entities will be *bona fide* ones and will therefore be accessible to subsequent presupposition triggers.

- **Conditionals**

As should be clear, a conditional is handled much like a conjunction: the antecedent can set up entities that can be picked up by presuppositions in the consequent. But where conditionals and conjunctions differ is that the entities created in conditionals may be hypothetical: they can be picked up by presuppositions in the rest of the utterance but cannot be picked up by presuppositions in subsequent utterances, and since they are variables will not license spurious entailments from the final logical form for the conditional.

To achieve this, I use a flag, called HYPO. HYPO is set by the constraint ⋆HYPO (hypothetical), which is in the lexical semantics of "if" (and also of "or" — see later). While the flag is set, the behaviour of the system does not really change: behaviour still depends on whether one or neither of the GIVEN and NEG flags is set. The only difference is that with the HYPO flag set, newly created entities have an extra property ascribed to them, also called HYPO. When sentence processing is done and information is moved from the ILC to the NIC, the change of data structure more clearly reveals entities labelled HYPO to be universally quantified variables. The lexical entry for "if" is:

word	*category*	*meaning*
"if"	$(S_x/S_y)/S_z$	CONDITIONAL(x) ANTECEDENT(x, z) ⋆HYPO(z) CONSEQUENT(x, y) ⋆HYPO(y)

I will illustrate conditional processing with example (8b) ("*If Launce owns a dog, his dog likes him*"), assuming that LAUNCE(L) is in the discourse model. We read in "if" and process its constraints. We create entities EV0, EV1 and EV2 for the satisfaction sets of x, y and z and label y and z as HYPO. "Launce owns a dog" is processed in normal fashion (creating D for the dog): we end up with OWNS(EV2), AGT(EV2, L), PAT(EV2, D) and DOG(D). Then "his dog likes him" is processed. We process "his dog" as usual and so pick up D as the referent. We end up with LIKE(EV1), AGT(EV1, D) and PAT(EV1, L). With processing done, we have the following:

```
CONDITIONAL(EV0)  ANTECEDENT(EV0, EV2)  HYPO(EV2)  CONSEQUENT(EV0, EV1)
HYPO(EV1)         LAUNCE(L)             OWNS(EV2)  AGT(EV2, L)
PAT(EV2, D)       DOG(D)                LIKE(EV1)  AGT(EV1, D)
PAT(EV1, L)
```

But the HYPO entities, EV1 and EV2, in fact have an interpretation as universally quantified variables. So in moving these constraints from the ILC to the NIC, we get:

$$\begin{array}{l}\mathtt{LAUNCE(L)} \wedge \forall ev_2 \forall ev_1 \forall d\,((\mathtt{OWNS}(ev_2) \wedge \mathtt{AGT}(ev_2,\ \mathtt{L}) \wedge \mathtt{PAT}(ev_2, d) \\ \wedge\ \mathtt{DOG}(d)) \supset (\mathtt{LIKES}(ev_1) \wedge \mathtt{AGT}(ev_1, d) \wedge \mathtt{PAT}(ev_1,\ \mathtt{L})))\end{array}$$

It might be questioned whether this universal reading is really what we want. It certainly is one reading of the utterance. There might be other possible readings (e.g. where the indefinite has a specific referent). I shall ignore this problem; it does not affect our consideration of presuppositions. What is most important is first that $\exists x\ \mathtt{DOG}(x)$ is not entailed by the final logical form, even though the presupposition of the consequent *was* satisfied, and secondly that the dog and the liking events are not accessible to subsequent presupposition triggers.

More complicated presupposition triggers would not cause any problems, e.g. "*If Launce hits the dog, he regrets he hit it.*": the treatment of factives given in Chapter 7 will handle a case like this correctly. The hitting will be HYPO and so, although the presupposition in the consequent can be satisfied, the hitting will not be entailed, nor will it be accessible in subsequent utterances.

There is a question about nesting conditionals within conditionals, and the accessibility constraints we might want to enforce with these cases. My intuitions suggest that, to the extent that this is allowed in English, T42 has enough generality to handle it. (To look at the question in DRT terms, it is the same as asking whether we need the capability to put boxes within boxes within boxes). The sorts of utterances under consideration here would be of the form, e.g., "If (if A then B) then C"[10] and "If A then (If B then C)"[11]. It would seem that as far as accessibility goes, we do not need anything more refined than I already have: later triggers in the conditional (no matter how complex it is) may access entities set up by earlier parts of the conditional.

Disjunctions

While it might be natural to consider now the behaviour of presuppositions in disjunctions, I shall postpone this discussion until Section 9.3.3, after I

[10] For example: "If whenever you see a play you feel bored, then we'll go to the cinema."

[11] For example: "If plays bore you, then provided you want to go out, we'll go to the cinema."

have looked at conditionals in more detail. This is because I will be arguing that disjunctions should, at least in some cases, be processed like conditionals. This means I will be taking disjunctions to be most usually *asymmetrical* with respect to presupposition projection, i.e. the ordering of material in an utterance, particularly its presupposition triggers, is important. A presupposition must be satisfied by something that came earlier (in this utterance or in previous discourse). Some people have argued that disjunctions are *symmetrical* with respect to presupposition projection, i.e. a presupposition of the first disjunct might be satisfiable by something that comes in the second disjunct. I will be considering the cases where this might be so. Before looking at this, a fuller understanding of conjunctions and conditionals will be useful.

Having considered the relevant features of T42 that enable it to handle presupposition projection, I will now give a comprehensive review of the data for coördinating constructions and indicate how T42 fares on this data.

9.3 Projection Problem Data for Coördinating Constructions

The most important factors involved in handling projection problem data for coördinating constructions are the order of the clauses in an utterance, the contents of the NIC prior to processing the utterance, and the updates made to the ILC by parts of the utterance that precede presupposition triggers.

I will remind you that I use * to label a syntactically or semantically anomalous example (in this section this usually is a self-contradictory utterance), I use ?? to label a pragmatically infelicitous utterance (in this section this usually means that there is a sense of redundancy in the utterance, e.g. when an utterance asserts what it previously presupposed), and I use ? to signify that the example needs further discussion, which I give in the subsequent text.

Again I embed (7a) ("*Launce's dog likes him*") into complex sentences (and again therefore sometimes use "*his dog likes him*" instead to get more natural-sounding examples).

9.3.1 Conjunctions

The easy cases for conjunctions are where there is a trigger in one conjunct but nothing related to it in the other:

(16) a. "*Launce is a good man and his dog likes him.*"
b. "*Launce's dog likes him and Launce is a good man.*"

In both of (16a) and (16b), the presuppositions are triggered, and whether the reading comes off depends simply on the presence in the NIC of, say, `DOG(Crab) ∧ OWNS(EV0) ∧ AGT(EV0, L) ∧ PAT(EV0, Crab) ∧ LAUNCE(L)`, i.e. a dog owned by Launce.

Now consider the cases where there is a direct match between a new object introduced in one conjunct and a presupposed referent in the other:

(17) a. "*Launce owns a dog and his dog likes him.*"
b. ??"*Launce's dog likes him and Launce owns a dog.*"

In (17a) it is unlikely that the NIC would contain a dog owned by Launce prior to utterance processing[12], since this is what the speaker is introducing in his first conjunct. This newly introduced dog can be used to satisfy the presupposition trigger in the second conjunct. This utterance thus places no special demands on the NIC.

In (17b), the trigger is in the first conjunct. It is encountered before the second conjunct has a chance to introduce a dog. Therefore, to satisfy the first conjunct's presupposition, the NIC must contain a dog owned by Launce prior to utterance processing. If it does not, the utterance reading will not come off. But suppose the NIC does contain a dog prior to utterance processing. In this case, the indefinite NP in the second conjunct will both create a new dog and manage to pick up the dog already in the NIC (see Chapter 6 on the processing of indefinite NPs). There is nothing to help it resolve this ambiguity (except perhaps a preference to choose readings that re-use existing entities, rather than ones that use new entities). However, beyond finding this unresolvably ambiguous in such a context, T42 does not really detect the redundancy in (17b).

I will now give some cases where one clause introduces an object and the other presupposes a similar object but there is no direct match in their descriptions:

(18) a. "*Launce owns an animal and his dog is playful.*"
b. ??"*Launce's dog is playful and he owns an animal.*"
c. "*Launce owns a puppy and his dog is playful.*"
d. ??"*Launce's dog is playful and he owns a puppy.*"

(18a) and (18c) can be felicitous in two ways. "His dog" may be coreferential with the animal or puppy introduced into the ILC by the indefinite NPs in their first conjuncts. In (18a) the indefinite NP is a more general descriptor and so further properties will be ascribed to the animal, namely that it is a dog. But alternatively, "his dog" could pick up a referent (if there is one) from the NIC and so refer to something completely different to the animal or puppy introduced in the first conjunct. Thus, conjunctions of this kind may exhibit 'intrasentential reference' (by picking a referent out of the ILC, where the referent is introduced by a previous clause of the utterance) or 'intersentential reference' (if there is an appropriate entity in the NIC). Many projection problem accounts do not recognise that these cases can have two readings: in one case demanding something of the non-immediate context and in the other case being satisfied by something internal to the utterance.

[12] Of course, throughout this chapter, if I say that the ILC or NIC must contain, e.g., a dog, this is an abridged form of saying that it must contain a discourse model entity which is predicated to be a dog, e.g. `DOG(D)`. The ILC and NIC contain formulas of logic, not 'real' dogs.

(18b) and (18d) have the presupposition trigger in their first conjuncts: if they are to be acceptable, a dog must be in the NIC before processing starts. The second conjunct can then either create a new animal or puppy, in which case Launce's animal or puppy would be quite different creatures to his dog, or the indefinite NPs can refer to Launce's dog. Again, as with (17b), T42 does not notice that there is redundancy in these utterances.

The examples in (19) are straightforward, but I include them because their corresponding conditionals will be less straightforward (see next subsection):

(19) a. "*Launce owns a dog and its tail wags*"
b. "*Launce has a grandchild and his child is happy.*"

In (19a), the first conjunct introduces a dog into the ILC. When this happens, T42 actions any forward-chaining inference rules in the manner described in Section 7.5.1. This adds the dog's tail to the ILC and this can then be referred to in the second conjunct. Similarly in (19b), the grandchild evokes a child entity, which can be presupposed in the consequent.

To conclude, I give some examples with negatives in them: I will only look at cases involving direct matches, similar to (17a) and (17b). All other cases are similar. Remember that negation, as described in Chapter 8, has narrow scope in T42 and so preserves presuppositions.

(20) a. *"*Launce doesn't own a dog and his dog likes him.*"
b. "*Launce owns a dog and his dog doesn't like him.*"
c. ??"*Launce's dog doesn't like him and Launce owns a dog.*"
d. *"*Launce's dog likes him and he doesn't own a dog.*"

In (20a), when T42 processes the first conjunct, it creates an owning event, `EV0`, and a dog, `D`, *in the scope of negation*. Thus the entities `EV0` and `D` are labelled `NEG`. Entities labelled `NEG` cannot be picked up by subsequent presupposition triggers, so for the trigger "*his dog*" to be satisfied in the second conjunct, since `D` cannot be used, there would have to be a dog owned by Launce already in the NIC. Given that the speaker's first conjunct asserts that this is not so, this presupposition will fail and the utterance reading will not come off.

(20b) is like (17a) and causes no problems. (20c) and (20d) have a presupposition trigger in their first conjuncts. If these utterances are to be acceptable, there must be a dog in the NIC. If there is, (20c) has redundancy, like (17b), and (20d) is self-contradictory.

I have now looked at examples showing no matches (16), exact matches (17), partial matches (18), inferred entities (19) and some cases with negatives (20): I believe this covers all the cases dealt with in the literature[13]. The approach I have described works equally well with other presupposition

[13] The more usual examples of partial match, for example, are: "*Mary has a son and her child is a philosopher*" and "*France has an intelligent king and the king of France is the only intelligent monarch in Europe*". These both correspond to example (18c).

triggers, such as factive verbs. Therefore this concludes the conjunction examples. Presupposition triggers pick up entities from earlier clauses or from the non-immediate context. At the end of utterance processing, we would, in all the acceptable cases, get the entailment $\exists x \exists y$(DOG(x) $\wedge$ OWNS(y) $\wedge$ AGT(y, L) $\wedge$ PAT(y,x)) (unless the conjunction is embedded within some other construct such as a conditional). In all these cases, the presupposition will have been satisfied and the entailment will therefore be possible.

9.3.2 Conditionals

The presuppositions of conditionals behave exactly as they do in the corresponding conjunctions. I rely on processing clauses in their order of occurrence and insist that every presupposition trigger be satisfied. The only difference is that newly created entities are labelled HYPO to make sure that they are regarded as variables in material implications. This ensures that the final logical form does not have spurious entailments.

Thus I claim that the data is exactly as it is in (16)-(20): where we have readings that do not come off there, we will get readings not coming off in conditionals too; where we are forced to find entities in the NIC there, we are forced to find them in the NIC for conditionals too; where we can link to an entity set up in the first conjunct (in the ILC) there, we can link to an entity set up in the antecedent in conditionals (and this might be a direct match or a partial match as was the case with conjunctions). It is in those cases where we do link to new entities set up in antecedents that we lose entailments.

Since there are such close parallels, I will look at only a handful of examples:

(21) a. *"If Launce is a good man, his dog likes him."*
b. *"If Launce's dog likes him, Launce is a good man."*
c. *"If Launce owns a dog, his dog likes him."*
d. ?*"If Launce's dog likes him, Launce owns a dog."*

For (21a) and (21b) to be acceptable, the NIC must supply an entity to satisfy the presupposition. Given that it does, then the final logical form will entail that Launce owns a dog. However, in (21c), "his dog" may pick up the dog introduced in the antecedent. Since this new dog is labelled HYPO, it translates as a universally quantified variable to give a final logical form that does not entail that Launce owns a dog:

$$\texttt{LAUNCE(L)} \wedge \forall ev_0 \forall d \forall ev_1 \, ((\texttt{OWNS}(ev_0) \wedge \texttt{AGT}(ev_0, \texttt{L}) \wedge \texttt{PAT}(ev_0, d) \wedge \texttt{DOG}(d)) \supset (\texttt{LIKES}(ev_1) \wedge \texttt{AGT}(ev_1, d) \wedge \texttt{PAT}(ev_1, \texttt{L})))$$

I have labelled (21d) with ? as it requires further comment. For (21d) to be acceptable, I am insisting that Launce's dog be in the NIC before utterance processing begins. In other words, I do not countenance the idea that the presupposition trigger in the antecedent could be satisfied by the dog entity introduced by the indefinite NP in the consequent. I maintain that a speaker would not use (21d) to introduce Launce's dog into the discourse model (as she might with "*If Launce owns a dog, ...*"): the dog must already be in the discourse model. If I am right, this is not a counterexample to the idea I have built into this theory of presupposition "projection", i.e. that presuppositions must be satisfied by things already in the discourse model and hence their satisfaction should be attempted in a strictly left-to-right order. If there is a dog in the discourse model, T42 will get two readings: in one it creates a new dog in the consequent, in the other it picks up the existing dog. Beyond this, as in (17b), T42 does not really detect the redundancy in this example.

But, there is a reason why (21d) does not read as 'unusually' as (17b) (??"*Launce's dog likes him and Launce owns a dog.*"). The reason is that there is another reading of (21d) (and possibly of all conditionals), but one which T42 does not detect. This other reading is due to conditionals being pragmatically ambiguous. Stalnaker characterises this ambiguity as follows:

> "If a person says something of the form 'If A then B' this may be interpreted either as the categorial assertion of a conditional proposition or as the assertion of the consequent made conditionally on the truth of the antecedent. In the former case, a proposition is determined on the level of semantics as a function of the propositions expressed by antecedent and consequent. In the latter case, the antecedent is an additional presupposition[14] made temporarily, either because the speaker wishes to commit himself to the consequent only should the antecedent be true, or because the assertion of the consequent would not be relevant unless the antecedent is true (as in, for example, "there are cookies in the cupboard if you want some")." [Stalnaker 1972, p.394].

T42 only computes the "categorial assertion of a conditional proposition", for which it uses material implication. The alternative reading is the case where "the speaker wishes to commit himself to the consequent only should the antecedent be true." In this case, the speaker is saying that she will accept that Launce owns a dog only by also accepting that the dog likes Launce: she perhaps cannot countenance Launce owning a dog that does not like him. Nevertheless, for the hearer to get even this reading, I am convinced that Launce's dog must have been previously introduced into the discourse model; (21d) alone cannot introduce this dog entity into the conversation.

In cases of partial matches, where a presupposition trigger ascribes extra properties to entities, some care has to be taken over the newly ascribed properties:

(22) "*If Launce owns an animal, then his dog is playful.*"

In (22), the presupposition trigger "his dog" can either be satisfied by the animal introduced into the ILC by the antecedent, or by a dog or other unspecified animal that was in the NIC before processing began. This is as

[14] "Presupposition" is not being used here in the sense I use it but rather in the sense of speaker presupposition, as described in Section 7.1.1.

it was with conjunctions, where we first noted this ambiguity. Either way, extra properties will be ascribed, namely that the animal is a dog. For the intrasentential reading, T42 puts these properties within the *antecedent* of the conditional, i.e.

$$\mathtt{LAUNCE(L)}\ \wedge \forall ev_0 \forall a\ ((\mathtt{OWNS}(ev_0) \wedge \mathtt{AGT}(ev_0,\ \mathtt{L}) \wedge \mathtt{PAT}(ev_0, a) \wedge \mathtt{ANIMAL}(a) \wedge \mathtt{DOG}(a)) \supset \mathtt{PLAYFUL}(a))$$

If, however, the animal was in the NIC beforehand, the newly ascribed property is not placed within the implication:

$$\mathtt{LAUNCE(L)} \wedge \mathtt{ANIMAL(A)} \wedge \mathtt{DOG(A)}\ \wedge \forall ev_0 \forall a\, ((\mathtt{OWNS}(ev_0) \wedge \mathtt{AGT}(ev_0,\ \mathtt{L}) \wedge \mathtt{PAT}(ev_0, a) \wedge \mathtt{ANIMAL}(a)) \supset \mathtt{PLAYFUL(A)})$$

Also needing extra discussion are cases where entities are inferred:

(23) a. “*If Launce owns a dog, its tail wags.*”
b. “*If Launce has a grandchild, his child is happy*”

The question is whether these inferred entities (i.e. the tail and the child) should be labelled HYPO or not, i.e. whether they will ultimately be added to the NIC as *bona fide* entities or whether they are variables. The problem is that intuitively in (23a) the tail that is inferred should be as hypothetical as the dog: it should be a variable:

$$\mathtt{LAUNCE(L)} \wedge \forall ev_0 \forall d \forall t \forall ev_1\, ((\mathtt{OWNS}(ev_0) \wedge \mathtt{AGT}(ev_0,\ \mathtt{L}) \wedge \mathtt{PAT}(ev_0, d) \wedge \mathtt{DOG}(d) \wedge \mathtt{TAIL}(t) \wedge \mathtt{HAS}(d,t)) \supset (\mathtt{WAGS}(ev_1) \wedge \mathtt{THEME}(ev_1, t)))$$

However, in (23b), the child that is inferred might be a *hypothetical* entity like the grandchild or it might be a *bona fide* entity. Presently, T42 only gets the reading where the child is hypothetical:

$$\mathtt{LAUNCE(L)} \wedge \forall ev_0 \forall g \forall c\, ((\mathtt{HAS}(ev_0) \wedge \mathtt{AGT}(ev_0,\ \mathtt{L}) \wedge \mathtt{PAT}(ev_0, g) \wedge \mathtt{GRANDCHILD}(g) \wedge \mathtt{CHILD}(c) \wedge \mathtt{BEGAT}(c, g)) \supset \mathtt{HAPPY}(c))$$

I believe that this is another case where plausible reasoning would be needed to license the other reading[15]:

$$\mathtt{LAUNCE(L)} \wedge \mathtt{CHILD(C)} \wedge \forall ev_0 \forall g\, ((\mathtt{HAS}(ev_0) \wedge \mathtt{AGT}(ev_0,\ \mathtt{L}) \wedge \mathtt{PAT}(ev_0, g) \wedge \mathtt{GRANDCHILD}(g) \wedge \mathtt{BEGAT}(\mathtt{C},\ g)) \supset \mathtt{HAPPY(C)})$$

The cases with negatives in conditionals also parallel the conjunction examples in (20) and so I will not repeat them here. However, so far we have only considered utterances of the form “If A, B”. We can also have utterances of the form “B, if A”. As it turns out, these can be handled using the same syntactic category for “if” as was given earlier. Again the examples parallel the conjunction examples: it is not the connective that matters, just the ordering of the presupposition triggers:

[15] The example often looked at in the literature is “*If John murdered his wife, he will be glad that she is dead.*”. The presupposition triggered by the factive “be glad that” will be satisfied by the death inferred from the murdering in the antecedent, but I believe this death may be hypothetical or *bona fide*. Which of these is the case should be determined by further plausible reasoning.

(24) a. *"Launce is a good man, if his dog likes him."*
b. *"Launce's dog likes him, if Launce is a good man."*
c. *"Launce owns a dog, if his dog likes him."*
d. ?*"Launce's dog likes him, if he owns a dog."*

Once more in (24a) and (24b) "his dog" must pick up an entity from the NIC, and in (24c) it can pick up the entity introduced in the first clause (irrespective of the fact that this is the consequent of the conditional). Again, this dog entity, which will be labelled HYPO, is a variable and will not give us a spurious entailment. In fact, (24c) is more likely to have the alternative reading licensed by Stalnaker's pragmatic ambiguity in conditionals. (24d) is like (21d): an entity must be in the NIC prior to processing. We must surely have been talking about Launce's dog beforehand, i.e. his dog must be in the NIC, if we are to utter (21d). It is not used to introduce a new dog. T42 will get two readings, one introducing a new HYPO entity and the other picking up the existing entity. Again we have Stalnaker's pragmatic ambiguity here.

9.3.3 Disjunctions

T42 does not at present handle disjunctions, so the following is a proposal rather than a report on work done, intended to show how disjunctions might be handled within the framework for processing compound utterances developed so far.

There would seem to be a problem with disjunctions, which can be illustrated with the following two examples:

(25) a **"Launce owns a dog or his dog likes him."*
b. *"Launce doesn't own a dog or his dog likes him."*[16]

In (25a) we do not want the presupposition trigger "his dog" to be able to pick up the dog introduced in the first disjunct. If this were to be possible, (25a) would have a reading. To prevent this, it would seem sensible that entities introduced in one disjunct should be inaccessible to presupposition triggers in the other disjunct. This seems a reasonable solution: the dog introduced in the first disjunct is not a *bona fide* entity since it is only hypothesised in a disjunction, not introduced in a simple assertion. It would also seem to be the approach one would use with Kamp's DRT: each disjunct would create a box and would not be able to access the contents of the other disjunct's box. Then the presupposition trigger in the second disjunct of (25a) will fail (unless there is already a dog in the NIC before utterance processing began, which, given that the speaker is using the disjunction to say that this need not be so, is unlikely).

As for (25b), the fact that the dog mentioned in the first disjunct is labelled NEG makes it inaccessible in the second disjunct, and if entities in disjuncts are inaccessible to each other anyway, it is 'doubly inaccessible'.

[16] One informant found this pragmatically infelicitous, but I do not believe it is.

This means that (25b) would be found unacceptable: it is again unlikely that there will be a dog in the NIC to satisfy the trigger in the second disjunct. But I consider (25b) to be have a reading. What seems to be needed here is for the presupposition to be able to pick up the dog mentioned in the first disjunct, even though this is contrary to what I have just said.

My proposed solution is very simple and yet I have not seen any precedent to it in the projection problem literature. I propose that we should make use of the fact that $(A \vee B)$ is equivalent to $(\neg A \supset B)$. I am not necessarily making any strong claims here about the equivalence or otherwise of natural language disjunctions and conditionals. The main thing I am offering is an idea about accessibility. Indeed, $(A \vee B)$ is also equivalent to $(\neg B \supset A)$, and we must consider what affect this has in explaining the data, and I will do this below. For now though, to give an idea of how this equivalence is supposed to work for us, I will continue with the equivalence of $(A \vee B)$ to $(\neg A \supset B)$.

The problem I have identified with disjunctions is that sometimes a presupposition trigger in the second disjunct should pick up its referent from the first disjunct and sometimes it should not. By reformulating a disjunction as a conditional according to the equivalence to $(\neg A \supset B)$, this problem resolves itself.

- A previously *positive* first disjunct (as in (25a)) becomes a negated antecedent. 'Entities' introduced within negatives are labelled `NEG`; they are existentially quantified variables within the scope of negation and cannot be used as anaphoric antecedents. This explains why the presupposition trigger in the second disjunct of (25a) cannot access the dog mentioned in the first disjunct, and hence explains why (25a) is probably infelicitous.

- A previously *negative* first disjunct (as in (25b)) becomes a doubly negative antecedent, which is to say it becomes a positive antecedent. Once it has become a positive clause, any entities it introduces are no longer within the scope of negation: they can be picked up to satisfy presuppositions. This explains how the presupposition trigger in the second disjunct of (25b) is allowed to pick up the dog mentioned in the first disjunct, and hence explains why (25b) has a reading.

The intuition here is that since cases such as (25b) are 'doubly inaccessible', this is like a double negation $(\neg\neg A \equiv A)$ and so they become accessible.

To explore this further, I will review a comprehensive set of examples and consider whether a reading for each is possible. In the following I give an example of a disjunction, e.g. (26a), and also the equivalent example in the form $(\neg A \supset B)$, in (26a′), and the equivalent in the form $(\neg B \supset A)$, in (26a″). In each example, the presupposition trigger is "Launce's dog" or sometimes "his dog". I indicate whether the translation demands an entity in the NIC before processing begins to satisfy the presupposition, or whether, it can be satisfied by something that comes earlier in the utterance, i.e. by something that would be in the ILC. Where a translation demands an entity in the NIC, I signal with a '?' those cases where I think it highly unlikely that there would be such an entity (and hence where it is unlikely that the presupposition would be satisfied), which is usually due to the speaker indicating elsewhere in the utterance that she is not assuming that the entity is in the NIC. Some of these cases labelled '?' might be more acceptable than others due to the pragmatic ambiguity of conditionals.

(26) a. "*Launce is a good man or his dog likes him.*"
a′. "*If Launce isn't a good man, his dog likes him.*" (NIC)
a″. "*If Launce's dog doesn't like him, Launce is a good man.*" (NIC)

b. "*Launce's dog likes him or Launce is a good man.*"
b′. "*If Launce's dog doesn't like him, Launce is a good man.*" (NIC)
b″. "*If Launce isn't a good man, Launce's dog likes him.*" (NIC)

c. ?"*Launce owns a dog or his dog likes him.*"
c′. ?"*If Launce doesn't own a dog, his dog likes him.*" (NIC)
c″. ?"*If Launce's dog doesn't like him, Launce owns a dog.*" (NIC)

d. ?"*Launce's dog likes him or Launce owns a dog.*"
d′. ?"*If Launce's dog doesn't like him, Launce owns a dog.*" (NIC)
d″. ?"*If Launce doesn't own a dog, his dog likes him.*" (NIC)

e. ?"*Launce owns a dog or his dog doesn't like him.*"
e′. ?"*If Launce doesn't own a dog, his dog doesn't like him.*" (NIC)
e″. ?"*If Launce's dog likes him, Launce owns a dog.*" (NIC)

f. ?"*Launce's dog doesn't like him or Launce owns a dog.*"
f′. ?"*If Launce's dog likes him, Launce owns a dog.*" (NIC)
f″. ?"*If Launce doesn't own a dog, his dog doesn't like him.*" (NIC)

g. "*Launce doesn't own a dog or his dog likes him.*"
g′. "*If Launce owns a dog, his dog likes him.*" (ILC)
g″. ?"*If Launce's dog doesn't like him, Launce doesn't own a dog.*" (NIC)

h. ?"*Launce's dog likes him or he doesn't own a dog.*"
h′. ?"*If Launce's dog doesn't like him, he doesn't own a dog.*" (NIC)
h″. '*If Launce owns a dog, his dog likes him.*" (ILC)

I will now draw some conclusions from this data. We can see that in the cases (26a-f) whether we treat $(A \vee B)$ as $(\neg A \supset B)$ or as $(\neg B \supset A)$ makes no difference. In all these cases the presuppositions can only be satisfied by there being a suitable entity in the NIC before processing begins. This is either because the presupposition trigger comes in the antecedent (26a″), (26b′), (26c″), (26d′), (26e″) and (26f′), or because the trigger is in the consequent but there is no suitable entity set up in the antecedent either because dogs are not mentioned in the antecedent (26a′) and (26b″), or because they are mentioned but are part of inaccessible negative existentially quantified expressions (26c′), (26d″), (26e′) and (26f″).

So, to repeat, in all of (26a-f), an entity must be in the NIC if we are to get a reading. This is quite possible for (26a) and (26b). However, it is unlikely in (26c-f) which contain clauses that explicitly assert the existence or non-existence of such an entity.

But now consider (26g) and (26h). In these, (26g″) and (26h′) are similar to the examples in (26c-f): the trigger is in the antecedent and so a dog entity must be in the NIC, but the consequent then explicitly asserts that Launce does not have a dog. However, (26g′) and (26h″) are perfectly acceptable. In these, the antecedent introduces a dog owned by Launce, which can be used to satisfy the presupposition in the consequent. What is interesting is that in all the examples (26a-f), whether we look at the $(\neg A \supset B)$ equivalence or the $(\neg B \supset A)$ equivalence, we still get the same predictions about the presuppositions, e.g. (26a′) and (26a″) give the same predictions. However, (26g′) and (26g″) give differing predictions, as do (26h′) and (26h″). Thus (26g) and (26h) are acceptable disjunctions (as shown in (26g′) and (26h″)), but perhaps we have some clue to why they feel slightly less acceptable than (26a) and (26b): it does not matter how you define accessibility for (26a) and (26b), but it does matter for (26g) and (26h).[17]

In fact, I find (26a) and (26b) perfectly acceptable, I find (26g) acceptable (although perhaps a bit of a struggle) and I do not like (26h) at all. This might be explained by the above data. (26a) and (26b) are acceptable whatever way we define accessibility. (26g) is acceptable on the equivalence given in (26g′), which is the $(\neg A \supset B)$ equivalence. This preserves the original surface ordering. Hence, accessibility here is given by the usual left-to-right processing that has worked in conjunctions, for example. However, (26h) is acceptable on the equivalence given in (26h″), which is the $(\neg B \supset A)$ equivalence. This does not preserve the original surface ordering. This might explain why (26h) is so awkward.

In fact, it would seem that (26h) can be improved if we start the sentence with "either": "*Either Launce's dog likes him or he doesn't own a dog.*". This is the only case where I find that "either" makes any difference to my intuitions.

This would seem to open the possibility to the following proposal. We should process English sentences of the form "A or B" asymmetrically with respect to presupposition projection. We can do this by giving them the accessibility constraints of $(\neg A \supset B)$. In the cases (26a-f) this makes no difference. In (26g) it gives us the reading that we want, but it rules (26h) to be unacceptable. We could then say that a sentence of the form "Either A or B" would have symmetric projection properties: this will give the same predictions for (26a-g) and will also admit a reading of (26h).

The attraction of this proposal, which claims that "A or B" differs from "Either A or B", are obvious when one is building an incremental NLP system which works from left-to-right through an utterance. It means that

[17] This might explain why one informant, as mentioned in the previous footnote, found (25b), which is the same as (26g), troublesome.

accessibility is signalled by the user: if she does not start her sentence with "either" we know that we can process straightforwardly from left-to-right and allow surface ordering to define accessibility. Only if she warns us, by starting with "either", will we need to process symmetrically. (While I believe that the data in (26) does go some way towards justifying this, I cannot honestly says that I am convinced that one should only get a reading for (26h) if it begins with the word "either").

If we do pursue such a solution, then the lexical entry for disjunction, i.e. for "or", for handling sentences of the form "A or B", would need to be something like the following:

word	*category*	*meaning*
"or"	$(S_x/S_y)/S_z$	CONDITIONAL(x) ANTECEDENT(x, z) ⋆HYPO(z) ⋆NEG2(z) CONSEQUENT(x, y) ⋆HYPO(y)

This is just like the entry for "if" except that it indicates that the antecedent must be negated. I have shown this with the NEG2 constraint. The problem will be to get NEG2 to work properly. What it must do is label all new constants introduced in the antecedent as NEG. If they are already labelled NEG, it must remove this label (removing a double negation). This must only be done to constants newly introduced in the first clause and really should only affect those that would fall under presupposition-preserving negation, i.e. from the verb to the end of the clause excluding other presupposition-triggering items. On reaching "or" in a sentence, how will T42 know which parts of the utterance that preceded the "or" would fall under the scope of negation?[18,19] If this can be done, then processing is exactly the same as for conditionals.

As for the processing for "Either A or B", if the system has a presupposition trigger which it cannot satisfy using the NIC, since the utterance began with "either", it knows that this trigger might yet be satisfied by something that comes later in the utterance. The sort of solution one could have here is like that given in Section 6.2 for the objects of verbs. For these, the satisfaction set is not evaluated; it is left as constraints until enough information to determine its extension is read in.

9.3.4 A Few Tricky Cases

In this subsection, I consider cases where the two clauses of a conjunction, conditional or disjunction share a *common* presupposition, and cases where the two clauses have *inconsistent* presuppositions.

First the case of a common presupposition:

[18] It cannot be every "new" thing that preceded the "or". In "*A man doesn't own a dog or his dog likes him.*", the dog in the first disjunct becomes accessible (since originally it is negated and then it gets negated again in transforming this into a conditional), and so the presupposition in the second disjunct will be satisfiable. But if the man introduced as a "new" entity in the first disjunct is also negated, he would become inaccessible and so no referent for "his" in the consequent would be found.

[19] This scheme is also highly inelegant. It has a certain non-monotonic feel to it as it is able to reach back and change the analysis of earlier parts of a sentence.

(27) a. *"Launce's dog likes Launce and Launce likes his dog."*
b. *"If Launce's dog likes Launce then Launce likes his dog."*
c. *"Launce's dog likes Launce or Launce likes his dog."*

These cause no trouble for T42's approach, and I mention them only because they do cause problems for some other approaches. In each of these examples, both clauses demand that a discourse model entity be found and this must be in (or inferable from) the NIC. If such entities can be found, the logical form for each of these will correctly entail that there is a dog owned by Launce.

Now I consider the cases of potentially inconsistent presuppositions. I have labelled all the examples ? because they all require further comment:

(28) a. ?*"Lear regrets he hit the fool and Lear regrets he didn't hit the fool."*
b. ?*"If Lear regrets he hit the fool, Lear regrets he didn't hit the fool."*
c. ?*"Lear regrets he hit the fool or Lear regrets he didn't hit the fool."*

(29) a. ?*"Launce met the King of Denmark and he met the President of Denmark."*
b. ?*"If Launce met the King of Denmark, he met the President of Denmark."*
c. ?*"Launce met the King of Denmark or he met the President of Denmark."*

It would appear that the presuppositions in (28) are incompatible and exhaustive, i.e. either Lear hit the fool or he did not, these two are inconsistent and exhaust the possibilities (disjunctively they form a tautology), whereas in (29), the presuppositions are incompatible but not exhaustive (disjunctively they do not form a tautology)[20], i.e. a country cannot have both a king and a president[21] but it could have neither if it has a queen.

The first thing to note is that it is not necessarily true that the inconsistency of the presuppositions would cause any problems. There *are* relatively straightforward readings for these utterances, but where a distinction is not drawn between the discourse model and possible worlds the readings I am about to explain cannot be obtained. The descriptive content of a presupposition in T42 is simply used to identify a discourse model entity, and its inconsistency with the descriptive content used to determine another entity need not impair the ability to satisfy both presuppositions. For example, in (28) it might be that the discourse model contains both an event in which Lear hit the fool and one (at a different time) in which Lear refrained from hitting the fool. Perhaps more obviously, in (29) both the presupposition demanding a King of Denmark and the one demanding a President of Denmark might be satisfied as long as the discourse model contains two entities, one which the conversational participants are referring to as the King and

[20] Mercer [1987] refers to similar examples as cases of "non-binary-valued features".

[21] This is a simplification since Albania has a president and an exiled king.

one which is being referred to as the President.[22] These readings are possible but perhaps not the usual ones. So it is necessary to consider what happens if there is only one discourse model entity being talked about in these utterances.

If there is only one entity which is a hitting of the fool by Lear, then the presuppositions in the first clauses in the examples in (28) will be satisfied, but the presuppositions in the second clauses will not be. Hence, in this situation, all of the examples in (28) would be found to be unacceptable. I believe this is correct. (Similarly if there is only an entity which is an event in which Lear refrained from hitting the fool, then, although the presuppositions in the second clauses will be satisfied, those in the first clauses will not, and again a reading will not come off). By the same argument, having only an entity that is a King in the discourse model will result in the examples in (29) being unacceptable as the demand for a President will not be satisfied (and *vice versa* if there is a President but no King entity).

What this does not explain is how it is that, intuitively, the disjunctive cases, (28c) and (29c) , would appear to be acceptable. It is necessary once again to consider the sort of contexts in which (28c) or (29c) might be uttered.

I believe (28c) would only be uttered in a context in which it has already been established that Lear and the fool had some sort of encounter (e.g. `EVENT(EVO)` $\wedge$ `AGT(EVO, Lear)` $\wedge$ `PAT(EVO, Fool)`). More than this is not known, e.g. the hearer does not know anything more about the exact nature of the encounter (i.e. there are no further properties known about `EVO`). In (28c), the speaker gives a couple of possible elaborations on the nature of the encounter which indicate that the encounter may have had the potential for Lear and the fool to come to blows. If I am right about this, then the discourse model should contain a *single* referent which could be picked up by the presupposition trigger in each of the clauses, i.e. the utterance does not demand two referents with conflicting properties, but one referent with rather underspecified properties. So there is a partial match between the first presupposition trigger and the discourse model entity representing this encounter. Additional properties are ascribed to the encounter saying that it was an encounter in which Lear hit the fool. These additional properties fall within the antecedent of the conditional in (28c′). The presupposition trigger in the second conjunct is also able to pick up the encounter entity and ascribe additional hypothetical properties to the entity to achieve its satisfaction. The fact that these additional properties fall within the material implication is important. If they were ascribed to the entity as *bona fide* properties (as they would be in (28a)) then there would be inconsistency and the reading would not be possible. It is only in conditionals and disjunctions where these properties are hypothetical that the reading is possible.

Again this may be clearer in (29). In (29c), an appropriate context would be one in which it has already been established that there is a constitutional

[22] This can easily happen in cases of inaccurate reference. Imagine a conversation in which, due to her lordly manner, we refer to Mrs. Thatcher as the President of Britain. A state visit to Britain by Launce might afford us the opportunity to say both that Launce met the Queen and that Launce met the "President" of Britain. (Happily, the latter encounter is now unlikely).

head of Denmark, i.e. `HEAD(H, Denmark)`, but the full identity of this person (`H`) is not known. Again the speaker provides some further specification: she suggests that `H` might have been the king or might have been the president; and again the presuppositions in the disjuncts are both satisfied by the same entity rather than by two entities with conflicting properties, but additional properties are ascribed to `H` within the conditional.

::::

In this chapter I have given T42's solution to the projection problem. There are one or two residual problems, but I believe the approach shows great promise. The approach is entirely uniform: presuppositions are always triggered and must always be satisfied by discourse model entities. The order of material in an utterance is very important: presuppositions must, with only one exception, be satisfied by entities introduced by previous discourse. It was suggested that the only counterexamples to this were those involving the possible symmetry of presupposition projection for disjunctions that begin with "either".

The advantage of the approach is that the mechanism is uniform and quite natural. There is no need to inhibit triggers or "suspend" or "cancel" presuppositions using any special mechanisms. These needs arise when presuppositions are not treated as preconditions. Although in T42 presuppositions are always triggered and satisfied by existing entities, this does not mean that these presuppositions will ultimately be entailed. Where an entity is introduced in a conditional (or disjunction), it is a universally quantified variable rather than a constant, and this loses the entailment.

One of the most important features of the approach is that it recognises ambiguity: inferred entities and extra properties ascribed to existing entities have, in certain cases, been added as *bona fide* entities and properties, and, in other cases, have fallen within the scope of material implication. We will see in the next chapter that most previous approaches have not catered for these ambiguities.

While disjunctions are not implemented in T42, I have been able to spell out the accessibility constraints for presuppositions in disjunctions by looking at the equivalence of logical disjunction with material implication, but this, and the cases of conflicting presuppositions in disjunctions, still require further work.

The next chapter reviews five main alternative approaches to presupposition projection. I show that each approach has problems with at least some of the data and that each approach has had to introduce some technical 'apparatus' to "cancel" presuppositions and is thus less simple than T42.

Chapter 10

Other Accounts of Presupposition Projection

In this chapter, alternative solutions to the projection problem are analysed. The proposals looked at are those of Karttunen & Peters (K&P), Gazdar, Mercer, Gunji and Van der Sandt (VdS). K&P is also reviewed in [Gazdar 1979, pp.108-119] and [Mercer 1987, pp.107-120], and Gazdar and Gunji are reviewed in [Mercer 1987, pp.120-149]. I do not know of any other reviews of Mercer's or VdS's work. I review K&P and Gazdar because they are the most significant linguistic approaches to presupposition projection. However, as they have been extensively reviewed elsewhere, I keep my reviews short. I spend more time on Mercer and Gunji because both are theories that were supported by computer implementations. Mercer's approach has not been reviewed elsewhere and its elegance makes it warrant attention here. Although Gunji has been reviewed in [Mercer 1987], Gunji's system exploits left-to-right processing in the same way as T42 does, and this justifies looking at the differences between Gunji's system and T42 here. Finally, VdS gives a linguistic theory which has many similarities with T42 in accounting for when a presupposition does and does not put demands on the non-immediate context.

All of the approaches reviewed here treat presuppositions as ordinary inferences. This means that in this chapter the word "presupposition" is not being used as it in my own approach. Treating presuppositions as ordinary inferences leads to the introduction of mechanisms for "cancelling" or "suspending" those presuppositions that should not be inherited by the utterance as a whole. Apart from the disadvantages inherent in having to use such mechanisms, none of the approaches except for VdS's can capture the ambiguous cases I characterised in Chapter 9.

Although all of the approaches treat presuppositions as ordinary inferences, they differ in the ways in which they achieve presupposition "suspension" or "cancellation". K&P do not so much cancel presuppositions as change them into other propositions (e.g. tautological ones). The other four approaches use some version of the idea of presupposition "cancellation": a potential presupposition (triggered in a constituent clause) will not survive if it is inconsistent with something else in the context. The precise details of this change from one approach to the next. None of the approaches is like my own, in which presuppositions are always triggered and must always be satisfied, but may be satisfied by "hypothetical" information, this giving the effects of presupposition "cancellation".

10.1 Karttunen & Peters

K&P's theory evolved through the 'Seventies [Karttunen 1973, 1974, K&P 1979]. I will only examine [K&P 1979] here. In this paper, K&P refer to presuppositions as "conventional implicatures". I think this is a misuse of Grice's term "conventional implicature" as originally conceived, and so will use the word "presupposition" in its place. I shall not go into K&P's formalism here but I will use the following notation: if A is a sentence of English, A^e are A's truth-conditions and A^p are A's presuppositions.

The basic approach is to divide subordinating expressions into "plugs" and "holes". Coördinating expressions are referred to as "filters" and have to be dealt with individually.[1] Presuppositions embedded under plugs, holes and filters are modified in different ways in order to "neutralise" them where appropriate.

10.1.1 Subordinating Constructions in K&P

Verbs which subcategorise for sentential complements are *plugs* if they do not allow presuppositions of their complements to be inherited. K&P's examples are verbs of saying such as "tell", "say" and "claim" and also external negation. However, we saw in Section 9.1.1 that presuppositions embedded under these verbs often *do* survive, i.e. K&P need to allow their plugs to leak, a phenomenon which I explained by an ambiguity in verbs of saying between quotational and non-quotational (leaky) uses. K&P recognise that there are such cases and say that, in such cases, the inferences that leak through the plug are not presuppositions but generalised conversational implicatures. They neither explain why such implicatures should arise, nor explain how it is that there is a plug which never leaks: external negation. It is definitional that K&P's external negation cancels presuppositions and that their internal negation does not (see Section 8.4).

A *hole* is a verb taking a sentential complement which does allow embedded presuppositions to be inherited. K&P's examples are "be glad that", "regret" and internal negation. This would seem to be correct. But K&P also claim that some holes *modify* a presupposition in inheriting it. Their examples are verbs of propositional attitude (e.g. "believe", "suspect" and "hope"): the presupposition is only inherited as a belief of the subject. For example, (1a) allegedly presupposes (1b), not (1c):

[1] By 1979, K&P were no longer using any of this terminology but they admit that their projection rules have the same effect as when this terminology was in place. I find it convenient to persist with the terminology.

(1) a. "*Launce believes that Proteus hit the dog.*"
b. 'Launce believes there is a dog.'
c. 'There is a dog.'

Modifying presuppositions in this way seems to be incorrect. For the negative "*Launce doesn't believe that Proteus hit the dog.*" (1b) need not hold true but (1c) should.

10.1.2 Coördinating Expressions in K&P

K&P refer to coördinating expressions as filters. The filters are A-and-B, If-A-then-B and A-or-B, where A and B are sentences. Each filter has its own treatment. I will look only at conditionals and disjunctions, where most of the problems arise[2].

If-A-then-B

- An utterance of a sentence of the form If-A-then-B presupposes $A^p \wedge (A^e \supset B^p)$. (In fact, the same formula is used for conjunctions). In the following (2a), (2c) and (2e) have presuppositions (2b), (2d) and (2f) respectively:

(2) a. "*If Launce is a good man, his dog likes him.*"
b. (Launce is a good man) $\supset$ (Launce owns a dog)
c. "*If Launce owns a dog, his dog likes him.*"
d. (Launce owns a dog) $\supset$ (Launce owns a dog)
e. "*If Launce owns a puppy, his dog is playful.*"
f. (Launce owns a puppy) $\supset$ (Launce owns a dog)

The projection formula seems to work for (2c) where it is not presupposed that Launce owns a dog. But intuitively (2a) should presuppose that Launce owns a dog, which it does not since it presupposes (2b). Similarly in (2e), while there is a reading in which it is not presupposed that Launce owns a dog (where the dog is the same animal as the hypothesised puppy), which we can take to be what (2f) conveys[3], there is another reading in which it should be presupposed that Launce owns a dog (different to the hypothesised puppy) which would be playful if Launce happened to own a puppy. This is not captured by K&P's formula.

To try to allow for the possibility of getting the right answer in (2a) and the alternative reading in (2e), K&P describe something they call *detachment*. By "detachment", it is possible for (2a) and (2e) to presuppose B^p, i.e. that Launce owns a dog. Their description of detachment is informal: this is the part of the paper not explained using logic. Mercer tries to tease out what K&P are saying (see [Mercer 1987, pp.113-120]). He says that the

[2] Fewer problems arise with conjunctions because presuppositions are also entailed in conjunctions (unless, for example, a conjunction is embedded in a conditional).

[3] Although (2f) does not show that the dog and the puppy are constrained to be the same animal under this reading.

only way B^p could be detachable from $A^p \wedge (A^e \supset B^p)$ is if A^e is known to be true either by being in, or by being entailed by, the context. But a speaker often utters a conditional when she believes that the truth of the antecedent, A, is *not* known.

A-or-*B*

- An utterance of a sentence of the form *A*-or-*B* presupposes $(A^e \vee B^p) \wedge (B^e \vee A^p)$. In the following (3a) presupposes (3b) which simplifies to (3c); (3d) presupposes (3e) which simplifies to (3f); (3g) also presupposes (3f):

(3) a. "*Launce is a good man or his dog likes him.*"
b. ((Launce is a good man) ∨ (Launce owns a dog))
∧ ((Launce's dog likes him) ∨ true)
c. (Launce is a good man) ∨ (Launce owns a dog)
d. "*Launce doesn't own a dog or his dog likes him.*"
e. ((Launce doesn't own a dog) ∨ (Launce owns a dog))
∧ ((Launce's dog likes him) ∨ true)
f. (Launce doesn't own a dog) ∨ (Launce owns a dog)
i.e. true
g. "*Launce's dog likes him or Launce doesn't own a dog.*"

I believe (3a) should presuppose that Launce owns a dog, rather than presupposing (3c). For (3d), it might (in a 'presuppositions as ordinary inferences' approach) be correct that (3d) loses any presuppositions (ending up with tautology (3f)). However, if the clauses in (3d) are reversed, as in (3g), since K&P's presupposition formula for disjunctions is symmetrical, a tautology would still be presupposed. This, I think, is wrong. Launce's dog *is* presupposed in (3g).[4]

A problem also arises with utterances which have presuppositions common to both disjuncts ((4a) presupposes (4b)) and utterances which have conflicting presuppositions in the disjuncts ((4c) presupposes (4d)):

(4) a. "*Launce likes his dog or his dog likes him.*"
b. ((Launce likes his dog) ∨ (Launce owns a dog)) ∧
((Launce's dog likes Launce) ∨ (Launce owns a dog))
c. "*Lear regrets he hit the fool or he regrets he didn't hit the fool.*"
d. ((Lear regrets he hit the fool) ∨ (he didn't hit the fool))
∧ ((Lear regrets he didn't hit the fool) ∨ (Lear hit the fool))

(4a) should presuppose that Launce owns a dog but it does not.[5] (4c) presupposes (4d), but (4d) is equivalent to the truth-conditions of (4c). Hence, (4c) presupposes itself, which is counter-intuitive.

[4] In earlier work, Karttunen [1973, 1974] used an asymmetrical formula of the form $A^p \wedge (\neg A^e \supset B^p)$. This formula still predicts that (3a) presupposes (3c), and (3d) presupposes (3f). But for (3g) it would presuppose that Launce owns a dog, in accord with my own intuitions.

[5] Soames [1979] revises K&P's filter for disjunction to $(A^e \vee B^p) \wedge (B^p \vee A^p) \wedge (A^p \vee B^p)$ so that the presupposition that Launce owns a dog in examples such as (4a) is obtained.

10.1.3 Final Remarks on K&P

Other Problems

We have seen that K&P can be criticised for not faring well on the projection data, and for resorting to informal explanations such as their notion of "detachment". The other major criticism levelled at K&P is that their classes (plugs, holes and filters) are not natural classes. They are *ad hoc*: they have no explanation except with regard to the data they purport to account for, i.e. you can only say which class something belongs to with reference to the data that the classes explain. Furthermore, the whole approach is based around processing compound sentences. It cannot handle stretches of text. It therefore cannot handle the cases of late cancellation mentioned in Chapter 7.

Weischedel

Weischedel based a computational system on K&P's work (in fact on Karttunen's earlier work [Karttunen 1973]) [Weischedel 1979]. His system used an ATN parser which also constructed a semantic representation and a presupposition set for a clause as it built the syntax tree. The presuppositions would be modified as the recursion of the ATN was unwound and it was realised what sort of embedding construction a presupposition had been triggered under.

Weischedel's projection algorithm was the same as Karttunen's except that verbs of saying were no longer straightforward plugs. For Weischedel, (5a) would presuppose (5b):

(5) a. "*Launce said that Proteus hit the dog.*"
b. 'Launce claimed there is a dog.'

This is wrong. It does not survive in presupposition-preserving negation ("*Launce didn't say that Proteus hit the dog.*"). Weischedel, like K&P, is ignoring the ambiguity of verbs of saying. In addition to making these wrong predictions, Weischedel's system suffers from all the other problems that K&P's theory does.

10.2 Gazdar

Gazdar's approach [Gazdar 1979] was probably the first to use the idea of explicitly cancelling 'potential presuppositions' if they were inconsistent with, in his case, clausal implicatures of the utterance and other contextual information.

10.2.1 The General Approach

Gazdar posits a set of functions which trigger inferences from the semantic representation of the constituent clauses of a compound utterance. These inferences are *potential* presuppositions and *potential* implicatures of the compound utterance; they will become *actual* presuppositions and *actual* implicatures provided they are not cancelled by Gazdar's projection algorithm. I will abbreviate potential presuppositions and implicatures by p-presuppositions and p-implicatures respectively[6].

The occurrence of definite NPs and factive verbs in an utterance trigger p-presuppositions. For example, (6a) triggers (6b) and (6c) triggers (6d):

(6) a. 'The Prince of Denmark smiled.'
b. 'K(There exists a Prince of Denmark)'
c. 'Lear regrets he hit the fool.'
d. 'K(Lear hit the fool)'

where K(A) means 'the speaker knows that A'. This raises a problem with utterances such as (7a) and (7b), which both presuppose (7c):

(7) a "*The bogeyman exists.*"
b. "*The bogeyman doesn't exist.*"
c. 'K(There exists a bogeyman)'

(7a) both asserts and presupposes (7c), i.e. it asserts and takes for granted the same proposition. On the other hand, (7b)'s presupposition, (7c), contradicts what (7b) asserts. Both of these situations are counterintuitive.

The p-implicatures that play an important role in presupposition projection are the generalised conversational clausal p-implicatures associated with conditionals and disjunctions. (8a) and (8b) both have clausal p-implicatures (8c)[7]:

(8) a. "If A then B."
b. "A or B."
c. P(A), P($\neg A$), P(B), P($\neg B$)

where P(A) means 'it is compatible with all the speaker knows that A' or 'for all the speaker knows, A'. Gazdar's notion of context is also important. He defines this as a set of consistent propositions to which the speaker is committed, i.e. a set of propositions of the form K(A).

Simple positive utterances also entail their p-presuppositions. So a simple positive utterance will both entail and presuppose the same information. This prevents cancellation of presuppositions in simple, positive utterances. On the other hand, since negation is vague (represented by wide-scope negation), negative utterances will not entail their p-presuppositions.

The processing of an utterance is as follows. First, all entailments, clausal p-implicatures and p-presuppositions for the utterance are determined. A compound utterance has, as its p-implicatures and p-presuppositions, *all* p-implicatures and p-presuppositions of its constituent sentences. Next, the

[6] Gazdar uses the abbreviations pre-suppositions and im-plicatures.

[7] Gazdar also deals with scalar p-implicatures, but as I have not found a case where they affect presupposition projection, I shall ignore them here.

context is incremented with these various inferences. However, the *order* of incrementation is important and something is only added to context if it is consistent for it to be added. If it is not consistent then that inference is cancelled (which simply means it does not get added). An inference can thus be cancelled either by something that was in the context before utterance processing began, or an inference can be cancelled by something which, by virtue of the ordered context incrementation, was put into the context just before an attempt to add the inference was made.

The order of incrementation is by type of inference: first entailments are added, then clausal p-implicatures and finally p-presuppositions. So entailments may cancel clausal p-implicatures. Those clausal p-implicatures which do not get cancelled are added to the context as *actual* clausal implicatures. Then we add the p-presuppositions, which may be cancelled by the entailments or the clausal implicatures that have just been successfully added. Those that are not cancelled are added as *actual* presuppositions.

It can be briefly demonstrated that the approach described above can handle cases of presupposition cancellation in negative utterances. (9a) has (9b) as its logical form and has (9c) as a p-presupposition:

(9) a. "*The Prince of Denmark didn't smile.*"
b. K(¬ (The Prince of Denmark smiled))
c. K(There is a Prince of Denmark)

(9b) is added to context first, followed by (9c) and, assuming there is nothing in context to the contrary, (9c) survives as a presupposition. However, (10a) has logical form (10b) and p-presupposition (9c):

(10) a. "*The Prince of Denmark didn't smile because there isn't a Prince of Denmark.*"
b. K(¬(The Prince of Denmark didn't smile) ∧ ¬(There is a Prince of Denmark))

(10b) is added to context first. (9c) cannot then be added because it is contradicted by the second conjunct in (10b): the p-presupposition is cancelled.

10.2.2 Compound Utterances in Gazdar

I will now look at some of the projection data for compound utterances. I will give two simple examples involving conditionals to illustrate how Gazdar's approach works and then look at some of the cases it fails on. In the examples, (a) is the utterance, (b) is the set of clausal p-implicatures, and (c) is the p-presupposition:

(11) a. "*If Launce is a good man, his dog likes him.*"
b. P(Launce is a good man), P(¬(Launce is a good man)), P(Launce's dog likes him), P(¬(Launce's dog likes him))
c. K(Launce owns a dog)

(12) a. "*If Launce owns a dog, his dog likes him.*"
b. P(Launce owns a dog), P(¬(Launce owns a dog)),
P(Launce's dog likes him), P(¬(Launce's dog likes him))
c. K(Launce owns a dog)

For (11a), assuming there is nothing already in the context that contradicts the inferences in (11b), the p-implicatures in (11b) are added and become actual implicatures. An attempt to add the p-presupposition (11c) is then made. The newly-added implicatures do not contradict (11c) hence (11c) may be added as a presupposition. Thus Gazdar correctly predicts the survival of this presupposition. This was one of the cases for which Karttunen & Peters needed to resort to their "detachment" argument (section 10.1.2). For (12a), again assuming the p-implicatures successfully become implicatures, (12c) cannot be added because it is cancelled: P(¬(Launce owns a dog)) is inconsistent with K(Launce owns a dog). Hence, (12c) is not a presupposition of (12a).

Gazdar's approach works well on most of the data, but there are some cases it fails on. The main problem is with cases involving only a partial match. (13a) has clausal p-implicatures (13b) and a p-presupposition (13c):

(13) a. "*If Launce owns a puppy, his dog is playful.*"
b. P(Launce owns a puppy), P(¬(Launce owns a puppy)),
P(Launce's dog is playful), P(¬(Launce's dog is playful))
c. K(Launce owns a dog)

Even if the p-implicatures become implicatures, the presupposition (13c) cannot be cancelled: the implicatures are not strong enough, i.e. P(¬(Launce owns a puppy)) is not inconsistent with K(Launce owns a dog). Hence, Gazdar predicts that the presupposition survives. Karttunen & Peters predict that it does not. I indicated in Chapter 9 that I believe one should be able to get both readings according to whether "his dog" in the consequent picks up as its referent a dog in the non-immediate context or the puppy introduced in the antecedent.

Another problem comes with conflicting p-presuppositions in a disjunction (see, e.g., [Mercer 1987]):

(14) a. "*Lear regrets he hit the fool or he regrets he didn't hit the fool.*"
b. P(Lear regrets he hit the fool), P(¬(Lear regrets he hit the fool)),
P(Lear regrets he didn't hit the fool), P(¬(Lear regrets he didn't hit the fool))
c. K(Lear hit the fool), K(¬(Lear hit the fool))

(15) a. "*Launce met the King of Denmark or he met the President of Denmark.*"
b. K(There is a King of Denmark), K(There is a President of Denmark)

For (14), the implicatures once added are not strong enough to cancel either of the p-presuppositions. Further, whichever of the two p-presuppositions gets added to the context first will cancel the other p-presupposition. Alternatively, if they are added simultaneously both will wrongly become presuppositions and the context will be inconsistent. Gazdar needs to modify his theory so that p-presuppositions can not only be cancelled by entailments and implicatures but so that mutually inconsistent p-presuppositions can cancel each other. If this were done, neither of the p-presuppositions in (14c) would survive, which, from a 'presuppositions as ordinary inferences' point of view, seems sensible.

But even this modification might not be enough. The p-presuppositions in (15) are mutually inconsistent but do not exhaust the possibilities. So even if they do cancel each other, as I proposed with (14), this may not be the 'right' answer. Mercer would say that a possible presupposition of (15a) (where presuppositions are viewed as ordinary inferences, and given that countries can have kings, presidents or queens) might be $\mathtt{K}$((There is a King of Denmark) $\vee$ (There is a President of Denmark)). Gazdar cannot obtain this.

10.2.3 Final Remarks on Gazdar

Other Problems

In addition to the problems Gazdar has with some of the data, the following criticisms have been levelled at Gazdar's 'solution':

- P-implicatures and p-presuppositions are technical devices introduced to make the theory work. Their status beyond this is not explained.

- It seems counterintuitive that presuppositions should be of the form $\mathtt{K}(A)$ rather than $\mathtt{Bel}(A)$. However, if $\mathtt{Bel}(A)$ were used, then p-presuppositions would not be cancelled by implicatures of the form $\mathtt{P}(A)$ and $\mathtt{P}(\neg A)$.

- As Gazdar himself notes, the order of context incrementation has no independent justification: entailments are added before p-implicatures, which are added before p-presuppositions, because things seem to work out properly this way.

- Once a p-presupposition becomes a presupposition, it can no longer be cancelled and yet I have indicated in Section 7.4 that there are cases where an utterance occurring later in a text may "cancel" a presupposition of an utterance that occurred earlier in the text.

Furthermore, Gazdar's approach is not adequately rescued by the various revisions that have been proposed as follows.

Landman

Landman [1981] attempts to derive stronger clausal implicatures for handling cases such as (13) where the implicature was too weak to cancel the p-presupposition. He proposes that entailments of clausal p-implicatures should be p-implicatures too. Assuming that a puppy is a young dog, (16a) then has the enlarged set of p-implicatures (16b) and the p-presupposition (16c):

(16) a. "*If Launce owns a puppy, his dog is playful.*"
b. P(Launce owns a puppy), P(¬(Launce owns a puppy)),
P(Launce owns a young thing), P(¬(Launce owns a young thing)),
P(Launce owns a dog), P(¬(Launce owns a dog)),
P(Launce's dog is playful), P(¬(Launce's dog is playful))
c. K(Launce owns a dog)

(16c) can now be cancelled. [Soames 1982] gives a number of reasons why this approach does not work more generally. The most important is that it now makes wrong predictions in cases which were previously correct. For example, (17a) will now have an enlarged set of p-implicatures, (17b), and a p-presupposition (17c):

(17) a. "*If Lear regrets he hit the fool, he is a humble man.*"
b. P(Lear regrets he hit the fool), P(¬(Lear regrets he hit the fool)),
P(Lear hit the fool), P(¬(Lear hit the fool)),
P(Lear is a humble man), P(¬(Lear is a humble man))
c. K(Lear hit the fool)

Now (17c) is cancelled when it should not be. A criticism of mine of Landman is that his attempts are misguided anyway: (16a) has *both* a reading in which "his dog" is the puppy (and so there is no entailment that Launce owns a dog) and one in which "his dog" picks up a referent from the non-immediate context (and there is an entailment).

Soames

Soames claims that many of the counter-examples to Karttunen & Peters' projection method are handled correctly by Gazdar's method, and that many of the counter-examples to Gazdar's method are handled correctly by K&P [Soames 1982]. Soames therefore attempted to synthesise the two approaches: he applies Gazdar's approach first and then passes the presuppositions this produces through K&P's filters. This synthesis now correctly handles (18), which caused K&P problems ((4c)):

(18) "*Lear regrets he hit the fool or he regrets he didn't hit the fool.*"

Soames first applies Gazdar's theory to the example. Assuming that inconsistent p-presuppositions can cancel each other, the p-presuppositions of the disjuncts are cancelled. There is thus nothing to submit to K&P's filters in the second part of Soames' synthesised theory. Thus Soames predicts that (18) has no presuppositions. This is correct.

However, Soames' synthesis does not resolve cases such as (15) ("*Launce met the King of Denmark or he met the President of Denmark.*"): the Gazdar part of Soames' theory gets no presupposition to submit to the K&P part of the theory and so we still cannot produce the presupposition K((There is a King of Denmark) $\vee$ (There is a Prince of Denmark)).

For cases of partial match as in (19a), application of Gazdar's approach leaves the presupposition (19b) (see (13)) but this is then submitted to K&P's filters $((A^p \wedge (A^e \supset B^p))$ to get (19c):

(19) a. *"If Launce owns a puppy, his dog likes him."*
b. 'Launce owns a dog'
c. (Launce owns a puppy) $\supset$ (Launce owns a dog)

Soames takes this to be correct: (19b) does not survive. But Soames, like K&P and Gazdar, fails to spot the ambiguity in these cases.

Soames' synthesis has all of Gazdar's problems (the unexplained order of context incrementation, the technical devices of p-implicatures and p-presuppositions, etc.) and has K&P's problems (the unnatural classes of plugs, holes and filters, for example) as well. It presumably also needs to have both K&P's internal and external negation to make the K&P part work *and* vague negation to make the Gazdar part work.

10.3 Mercer

Mercer's work [Mercer 1987] can be seen as one particular computer implementation of Gazdar's theory because, following Gazdar, Mercer has based his solution on the notion of consistency with context. However, by using a default logic approach, Mercer avoids Gazdar's unexplained technical 'apparatus', e.g. the notion of 'potential presuppositions' and the ordered incrementation of context, and fares slightly better on the data.

10.3.1 The General Approach

For Mercer, positive utterances do not have presuppositions. The "presuppositions" of simple, positive sentences are simply entailments. While this explains why these are not cancellable, it fails to give any explanation of their 'background feel'. Negative utterances are represented using vague (wide-scope) negation. Non-monotonic, default inference rules might apply to license the drawing of presuppositions from such utterances. Since these are default inferences, information already in context or new, incoming information might cause their inhibition or retraction.

The default logic is based on that in [Reiter 1980]. Mercer is able to restrict himself to "normal default rules", which are of the form:

$$\alpha(\vec{x}) : \beta(\vec{x})/\beta(\vec{x})$$

where $\alpha(\vec{x})$ and $\beta(\vec{x})$ are first-order formulae whose free variables are among those of $(\vec{x}) = x_1, \ldots, x_m$. α is the *prerequisite*, the first occurrence of β is the *justification* and the second occurrence of β is the *consequent*. This rule can be read as: for all individuals $x_1, \ldots, x_m$, if $\alpha(\vec{x})$ is believed and if $\beta(\vec{x})$ is consistent with the current set of beliefs, then conclude $\beta(\vec{x})$. I shall not go into the details of the model-theory or proof-theory of this logic; instead I shall rely on an intuitive understanding of it. Roughly, $\beta(\vec{x})$ can be concluded if $\alpha(\vec{x})$ can be proved to be true and the knowledge base does not contradict $\beta(\vec{x})$.[8]

To make this more concrete, I will look at the inference rules for factive verbs:

(20) Axiom schema: $\forall x(P(x,\Phi)\wedge \texttt{FACTIVE}(P) \supset \Phi)$
Default rule schema: $\neg P(x,\Phi)\wedge \texttt{LF}(P,x,\Phi) \wedge \texttt{FACTIVE}(P) : \Phi/\Phi$

The axiom schema says that if x P's Φ where P is a factive, then Φ is true. This deals with the entailment of the complement of a factive in a simple, positive utterance. The default rule schema says that if it is not the case that x P's Φ where P is a factive, and Φ is consistent with what is already known, then Φ is true. In other words, in a negative utterance, the complement Φ can be concluded if it is consistent so to conclude it.

The default rule schema also contains the predicate LF. The predicate LF is there simply "... to guard against misuse of the default rules." [*ibid.*, pp.72-73]. Default presupposition rules should only be triggered if there is a presupposition trigger in the original utterance, i.e. presuppositions are only triggered by presupposition triggers and these triggers can only occur in natural language utterances. Presupposition default rules should not be fired by bits of logic generated by some other process, only by bits of logical form that originated from translation of a natural language utterance into logic. To ensure that this will be so, bits of logical form originating from utterance translation are labelled LF and presupposition default rules as in (20) have additional preconditions that they apply only to things labelled LF. The LF predicates can occur only in these two places hence ensuring the linkage. Mercer recognises that this is inelegant and that perhaps greater integration of presupposition default triggering and utterance translation could help solve this problem.[9]

A particular example would be (21a), whose semantic representation (following [Mercer 1987, item (3.61), p.79]) is (21b):

(21) a. *"Lear does not regret he hit the fool."*
b. K(¬ REGRET(Lear, HIT(Lear, Fool))
∧ LF(REGRET, Lear, HIT(Lear, F)))

On adding (21b) to the knowledge base, the default rule in (20) can be instantiated to give ¬REGRET(Lear, HIT(Lear, Fool)) ∧ LF(REGRET, Lear, HIT(Lear, Fool)) ∧ FACTIVE(REGRET) : HIT(Lear, Fool) / HIT(Lear, Fool). The prerequisites of this are true and, assuming the knowledge base indicates that "regret" is a factive verb, i.e. FACTIVE(REGRET), and that the justification is consistent with other things in the knowledge base, then

[8] I will ignore the problem of this logic, that is, that its proof-theory is undecidable.

[9] This might be an oblique call for an incremental system! LF is only required because types of processing are not properly interleaved.

the consequent may be concluded. This is the presupposition, HIT(Lear, Fool).

Now consider (22a) and its logical form (22b):

(22) a. "*Lear doesn't regret he hit the fool because he didn't hit the fool.*"
b. K(¬ REGRET(Lear, HIT(Lear, Fool)) ∧
LF(REGRET, Lear, HIT(Lear, Fool)) ∧¬ (HIT(Lear, Fool)))

Since ¬(HIT(Lear, Fool)) is entailed by (22b), the default rule's justification is contradicted so the consequent (presupposition) will not be concluded (i.e. will be correctly "cancelled").

Mercer has no problem with late cancellation, i.e. utterances cancelling presuppositions set up by earlier parts of a text. Technically presuppositions are not added to the theory they are generated from: they are part of an extension to the theory. If a new clause is added to the theory, no matter how much later this is done, the extension(s) are recomputed from scratch. This time any "cancelled" presuppositions would not be in the extension.[10]

It is interesting that in the representations (21b) and (22b) Mercer has assumed that definite NPs such as "Lear" and "the fool" can pick up referents (constants such as Lear and Fool) and plug them into the logical form. Since these are constants, they can no longer fall under the scope of the negation, i.e. the representation is not able to represent "*Lear doesn't regret he hit the fool because there is no fool.*" Indeed, nowhere does Mercer apply his approach to definite NPs. Why this is he never says. It may be the difficulty of capturing the trigger of a definite NP in his logic. I do not know. It means that Mercer has one (incomplete) mechanism for definite NPs and a different mechanism for other types of presupposition. T42's approach has the virtue of giving a uniform account of the different cases of presupposition.

10.3.2 Compound Utterances in Mercer

To handle compound utterances, Mercer does case analyses on their clausal implicatures.[11] An inference will be an *entailment* of the compound if it is entailed in all the cases. An inference will be a *presupposition* of the compound if it is inferable in all the cases but in at least one of these cases it is derived using a default rule. An inference will not be an inference of the compound if there is a case in which it is not inferable.

I will now show how Mercer handles the data. Since it is not clear how he would handle definite NPs, I will have to use different examples to those used in previous sections. I will use examples with factives in them. (23a)'s logical form, (23b), and its clausal implicatures, (23c), would be added to a

[10] Mercer recognises the need to use a Reason Maintenance System to handle retractions without full recomputation of the theory's extensions.

[11] Although Mercer presents this as a case analysis, he points out that for technical reasons this is not quite so. For a default rule to fire, its prerequisites must be provable, but in a case analysis, for a default rule to fire, its prerequisites need be provable only in that case. The default rules can be rewritten to overcome this problem.

theory containing an axiom, (23d), a default rule, (23e), and the knowledge that "regret" is factive, (23f):

(23) a. "*If Lear is sane, he regrets he hit the fool.*"

b. $\texttt{K(SANE(Lear)} \supset \texttt{REGRET(Lear, HIT(Lear, Fool)))}$
$\wedge\ \texttt{LF(REGRET, Lear, HIT(Lear, Fool))}$[12]

c. $\texttt{P(SANE(Lear)), P(}\neg\texttt{SANE(Lear)),}$
$\texttt{P(REGRET(Lear, HIT(Lear, Fool))),}$
$\texttt{P(}\neg\texttt{REGRET(Lear, HIT(Lear, Fool)))}$

d. $\forall x\texttt{(REGRET(}x\texttt{, HIT(Lear, Fool))} \wedge \texttt{FACTIVE(REGRET)}$
$\supset \texttt{HIT(Lear, Fool)}$

e. $\neg\texttt{REGRET(}x\texttt{, HIT(Lear, Fool))}$
$\wedge\ \texttt{LF(REGRET, }x\texttt{, HIT(Lear, Fool))} \wedge \texttt{FACTIVE(REGRET) :}$
$\texttt{HIT(Lear, Fool) / HIT(Lear, Fool)}$

f. $\texttt{FACTIVE(REGRET)}$

The cases of the case analysis are (24a) and (24b):

(24) a. $\texttt{P(}\neg\texttt{SANE(Lear)} \wedge \neg\ \texttt{REGRET(Lear, HIT(Lear, Fool)))}$
b. $\texttt{P(SANE(Lear)} \wedge \texttt{REGRET(Lear, HIT(Lear, Fool)))}$

In case 1, (24a), (25a) is assumed; (25b) is an entailment of (25a) using ∧-elimination. (25c) can be derived using the default rule (23e) since there is nothing to the contrary.

(25) a. $\neg\texttt{SANE(Lear)} \wedge \neg\ \texttt{REGRET(Lear, HIT(Lear, Fool))}$
b. $\neg\texttt{SANE(Lear)}$
c. $\texttt{HIT(Lear, Fool)}$

In case 2, (24b), (26a) is assumed; (26b) is an entailment of (26a) by ∧-elimination. (26c) is derivable using the axiom (23d):

(26) a. $\texttt{SANE(Lear)} \wedge \texttt{REGRET(Lear, HIT(Lear, Fool))}$
b. $\texttt{SANE(Lear)}$
c. $\texttt{HIT(Lear, Fool)}$

Since $\texttt{HIT(Lear, Fool)}$ is derived in both cases ((25c) and (26c)), it is an inference of the original utterance, but as it is an entailment in one case and a default inference in the other, overall it has presuppositional status. This is correct.

My presentation will be much more sketchy with subsequent examples. For (27a), the cases are (27b) and (27c):

(27) a. "*If Lear hit the fool, he regrets he hit the fool.*"
b. $\texttt{P(}\neg\texttt{HIT(Lear, Fool)} \wedge \neg\ \texttt{REGRET(Lear, HIT(Lear, Fool)))}$
c. $\texttt{P(HIT(Lear, Fool)} \wedge \texttt{REGRET(Lear, HIT(Lear, Fool)))}$

[12] I use Mercer's logical forms here: they are higher-order since $\texttt{REGRET}$ is taking a propositional argument. He says they can be thought of as abbreviations for longer first order forms using event variables.

In case 1, (27b), ¬HIT(Lear, Fool) is an entailment and hence the default rule is not triggered. In case 2, (27c), HIT(Lear, Fool) is an entailment. Combining the two cases gives ¬HIT(Lear, Fool) ∨ HIT(Lear, Fool) which is a tautology. Hence, correctly, there is no substantive presupposition.

Mercer does not properly consider cases of partial match (although see the discussion below), so my treatment below is putative. Consider (28a), where punching is more specific than hitting and so the hitting might refer to the punching or to some other hitting. The cases are (28b) and (28c).

(28) a. "*If Lear punched the fool, he regrets he hit the fool.*"
b. P(¬PUNCHED(Lear, Fool)
∧¬ REGRET(Lear, HIT(Lear, Fool)))
c. P(PUNCHED(Lear, Fool) ∧ REGRET(Lear, HIT(Lear, Fool)))

In case 1, (28b), ¬PUNCHED(Lear, Fool) is an entailment. This is not strong enough to inhibit the default rule (i.e. just because Lear did not punch the fool does not mean that Lear did not hit the fool) and so the default rule will fire and HIT(Lear, Fool) will be obtained as a default inference. In case 2, (28c), PUNCHED(Lear, Fool) and HIT(Lear, Fool) are entailments. Since HIT(Lear, Fool) is an inference in each case but is an entailment in one and a default inference in the other, HIT(Lear, Fool) is a presupposition of (28a). But this is only one reading. Mercer, like Gazdar, is unable to get the reading where I would have the presupposition being satisfied by the more specific referent in the antecedent.

As mentioned above, Mercer does not look at examples with hitting and punching (or dogs and puppies). But he does look at (29a). He gets presupposition (29b) from the cleft construction in the consequent of (29b):

(29) a. "*If someone at the conference solved the problem, it was Julian who solved it.*"
b. 'Someone solved the problem.'

To get the alternative reading, where this is not presupposed, he claims that the consequent should be treated as having had something deleted from it. The full utterance is to be thought of as (30), he claims, and this would not have the presupposition:

(30) "*If someone at the conference solved the problem, then of those at the conference it was Julian who solved it.*"

In other words, Mercer recognises that the antecedent in (29a) is too weak to cancel the presupposition (29b), but instead of trying to make the (implicatures of the) antecedent stronger as Landman did, he makes the presupposition of the consequent weaker by adding extra information into the consequent. This should make the p-presupposition more specific and hence susceptible to cancellation. Mercer gives a number of justifications for inserting the extra text into the consequent, the most interesting of which is that he suggests that the cleft seems to refer back to the event mentioned in the antecedent clause. This shows that he and I are thinking along the same lines here. One difference is that T42 handles this automatically and Mercer's system does not. Another difference is that while Mercer can thus account for example (29), it is not clear that any simple insertion of 'missing' text will account for examples such as (28): his account lacks generality.

However, Mercer's account is good at handling the cases of conflicting presuppositions such as in (31) and (32)[13]:

(31) a. "*Lear regrets he hit the fool or Lear regrets he didn't hit the fool.*"
b. P(REGRET(Lear, HIT(Lear, Fool))
∧¬ REGRET(Lear, ¬ HIT(Lear, Fool)))
c. P(¬ REGRET(Lear, HIT(Lear, Fool))
∧ REGRET(Lear, ¬ HIT(Lear, Fool)))

(32) a. "*Othello regrets he is white or Othello regrets he is black.*"
b. P(REGRET(Othello, WHITE(Othello))
∧¬ REGRET(Othello, BLACK(Othello)))
c. P(¬ REGRET(Othello, WHITE(Othello))
∧ REGRET(Othello, BLACK(Othello)))

For (31a), in case 1, (31b), HIT(Lear, Fool) is entailed which inhibits the default rule so ¬ HIT(Lear, Fool) is not a default inference. In case 2, (31c), ¬ HIT(Lear, Fool) is entailed which again inhibits the default rule. Combining the cases gives HIT(Lear, Fool) ∨¬ HIT(Lear, Fool) which is a tautology: there is no substantive presupposition. This is reasonable in a 'presuppositions as ordinary inferences' account.

For (32a), the cases are (32b) and (32c). Following an example in [Mercer 1987, pp.128-130] (but simplifying it a bit), I will assume that people in a southern African regime are classified by law as either blacks, whites or coloureds. In case 1, (32b), WHITE(Othello) is entailed which inhibits the default rule so BLACK(Othello) is not a default inference. In case 2, (32c), BLACK(Othello) is entailed which inhibits the default rule so WHITE(Othello) is not a default inference. Combining the cases gives WHITE(Othello) ∨ BLACK(Othello). This is not a tautology since people can also be coloured. Hence the presupposition is WHITE(Othello) ∨ BLACK(Othello) which appears reasonable.

To summarise, Mercer appears to be able to handle all the data including the tricky disjunction cases. The exceptions are the cases of partial matching where Mercer's framework at present only gets one reading. A possible further counterexample occurs in cases such as (33a) and (33b). Mercer and I disagree about these. I suggested they do presuppose (33c) and are only sensible owing to the pragmatic ambiguity of conditionals. Mercer, however, says they do not presuppose (33c). He says in case 1, (33d), ¬ HIT(Lear, Fool) is an entailment which inhibits the default rule. In case 2, (33e), HIT(Lear, Fool) is an entailment. Combining the cases gives ¬ HIT(Lear,

[13] Again I cannot use my usual examples of Launce meeting the King or the President because Mercer does not deal with definite NPs.

Fool) ∨ HIT(Lear, Fool) which is a tautology. Mercer, therefore, predicts no substantive presupposition.

(33) a. "*If Lear regrets he hit the fool, he hit the fool.*"
b. "*Lear regrets he hit the fool, if he hit the fool.*"
c. HIT(Lear, Fool)
d. P(¬ REGRET(Lear, HIT(Lear, Fool))
∧¬ HIT(Lear, Fool))
e. P(REGRET(Lear, HIT(Lear, Fool)) ∧ HIT(Lear, Fool))

10.3.3 Final Remarks on Mercer

Mercer's scheme is elegant. It has no *ad hoc* apparatus (no unnatural classes such as plugs and holes, no p-presuppositions, no unexplained order of context incrementation, etc.), although there are computational problems with default logic (e.g. its undecidability).

But it is an approach based on ordinary inferences, which I disagree with, and there are also a number of putative counterexamples to the scheme. However, its greatest shortcoming is that it is not fully integrated with utterance processing. This gives rise to a number of problems. First, Mercer cannot *use* presuppositions to resolve ambiguity. Secondly, it leads to the lack of uniformity between Mercer's processing of definite NPs and other presupposition triggers. I find it strange that Mercer does not deal with definite NPs. It means that definite NPs, which have up to now been the archetypical presuppositional phenomena, have a different analysis in his system to other presupposition triggers: definite NPs retrieve referents whereas factives give rise to a complex logical form and have default rules applied to them. I would regard it a success of T42 over Mercer that I have shown how all these cases can have a uniform treatment in terms of retrieving referents from a discourse model. Thirdly, it means he cannot handle the cases where indefinite NPs occurring in presuppositional environments behave anaphorically (e.g. "*Bassanio regrets he made a deal.*" and "*If Dogberry came to the party, then the hostess must have been really glad that there was a policeman present.*") (see Section 7.3.2).

10.4 Gunji

Gunji's approach [Gunji 1981], like Mercer's, can be viewed as an attempted computer implementation of Gazdar's theory. Gunji's system is much less successful on the data than Mercer's and still needs analogues to some of Gazdar's technical 'apparatus', e.g. the idea of potential presuppositions. However, Gunji's work is interesting for the way it tries to solve the problem relying in part on processing utterances from left-to-right. As we saw in Chapter 9, this is a very important part of T42's solution to the projection problem.

10.4.1 Gunji's System

Gunji describes an unusual architecture for NLP which integrates semantic and pragmatic processing. He describes his approach as "model-theoretic". For an input sentence, it produces a formula of his meaning representation, Extended Intensional Lisp (EIL), which is evaluated against some representation of a model of the world (i.e. evaluated against a database[14]). This evaluation both returns some sort of denotation, e.g. the denotation of a declarative sentence would be a truth-value, that of a noun phrase would be the noun phrase's referent(s), and may also update the model. By returning denotations it supposedly handles compositional semantics, and by updating the model it supposedly handles pragmatics.

EIL expressions are evaluated sequentially, from left-to-right. Gunji makes allusions to this being psycholinguistically plausible but his prime motive for this order of evaluation is, like my own, that certain pragmatic phenomena seem to be more naturally handled this way: given that evaluation of one expression might update the model, evaluation of a subsequent expression may occur in a new environment. Gunji says that as the model is capable of being updated it can, in some sense, only be a partial model. This is unlike more usual model-theoretic semantics where the model is thought of as complete. It is not clear what Gunji means by "partial" and, as I shall explain later, I believe this claim to be bogus.

The system has the following components: a *syntactic parser*, which maps sentences to expressions of EIL; a *translator*, which takes the EIL formula and recursively evaluates its semantic and pragmatic procedures against the model; and a *super-interpreter*, which takes the denotation of the sentence and determines some further action to take such as changing the model or returning an answer to the user based on the sentence-type (declarative, interrogative, etc.).

Gunji says that the parse tree which the parser produces is one "...where quantifier scopes are disambiguated, all the categories of lexical items are identified so that their functional relations are explicit, some of the pronoun references are determined, ellipses are recovered, etc." [Gunji 1981, p.3]. Nothing more is said about the operation of the parser, except that Gunji says in a footnote: "The parser may have to be more closely knit with the interpreter and the super-interpreter than it is assumed to be here in order to resolve ambiguities, e.g. different scope relations based on the context. Since the parser itself can only list possible analysis trees, the determination or preference of a particular tree must be based on the information 'fed back' from the interpreter and the super-interpreter." [*ibid.*, footnote 1, p.4]. Thus, he seems to recognise the need for an incremental system, although he has not produced one.

Before I describe Gunji's semantic interpretation process, I will describe the model a little more. The model is a database whose relations say which (tuples of) individuals do and do not satisfy a predicate[15]. For example, an

[14] Gunji uses the terms "database" and "model" interchangeably for his system.

[15] Since EIL is an intensional and tense logic the model must include the set of possible

EIL predicate such as `bald` will have what Gunji calls a *d-list* holding those individuals who are known to be bald and an *nd-list* holding those individuals known to be not-bald, i.e. a true-list and a false-list. The model is "partial" so individuals need not appear on either list. For such individuals, a query to the database about their baldness would not return true (`T`) or false (`F`) but undefined (`U`).

The database is organised into a set of hierarchical "universes of discourse" (UDs): $\mathrm{UD}_0, \mathrm{UD}_1, \ldots, \mathrm{UD}_n$, where UD_i ($n \geq i > 0$) contains information invoked by the ith utterance in the discourse, and UD_0 supposedly holds general world knowledge, although the nature of this is not described. UDs provide a focussing mechanism: arranging UDs in this way means that searching from the most recent UD_n to UD_0 finds the more focussed elements first.[16]

To present Gunji's evaluation procedures in detail would be inappropriate. Instead, I will give two simple examples to show how the system works. The examples I use are (34a) and (35b) ((a) is the sentence and (b) is its translation in EIL):

(34) a. "Prospero has a car."
b. `(some(x) (and (car x) (have P x)))`[17,18]

(35) a. "The car is black."
b. `(some (x) (and (car x) (every (y) (and (car y) (equ x y))) (black x)))`[19]

I will describe processing of (34b) first. Evaluation of `some` returns the result of evaluating `(and (car x) (have P x))`. This involves evaluation of `and`. `and` returns true if its arguments evaluate to sets with a non-null intersection. Evaluation of the argument `(car x)` returns a list of cars known to the system and then evaluation of the other argument, `(have P x)`, limits this to those cars that `P` has. Thus (34b) is basically interpreted as a search of the UDs for a satisfying instance, i.e. an entity that instantiates `x` that is both a car and owned by Prospero. If there is a satisfying instance then T (true) is returned to the super-interpreter. (35a) receives a similar treatment to (34a), but there is an an added uniqueness constraint.

But there is a problem with this evaluation procedure. Suppose there was no satisfying instance for (34b), and so evaluation returned U (undefined).

worlds and the set of moments of time. Gunji's description of these and the role they play in handling modal and tensed expressions is cursory so I ignore them here. They are not apposite to the points I wish to bring out.

[16] It is not clear that this makes much difference to system behaviour as searches appear to be exhaustive anyway. Furthermore, this focussing is too primitive. It is based at the sentence level. For "Prospero gave Caliban a book", Prospero, Caliban and the book will all be in the same UD: a more refined level of focussing is needed.

[17] These are lexical items of EIL. Despite superficial similarity, these are not the predicates and quantifiers of first-order predicate calculus. Nor is `car` the Lisp function `car`, it is an Extended Intensional Lisp function denoting automobiles.

[18] Note that Gunji assumes that his *parser* can determine that Prospero has referent `P`.

[19] This is not in fact the EIL expression Gunji gives. He gives `(some (x) (and (every (y) (equ (car y) (equ x y))) (black x))`. This must be wrong: even the brackets do not match. The first `equ` must be a mistyped `and`.

There is no place in Gunji's evaluation functions that permits the creation of new entities, even for indefinite noun phrases. This means that the set of entities in Gunji's model must be a complete set. Not only is this counter-intuitive, it also brings into question in what sense Gunji's model is "partial". It must be that the set of entities in the model is complete but our knowledge about their properties is not. So, for example, if Prospero is (known to be) bald, he will appear on the *d-list* of `bald`; if he is (known to be) not-bald, he appears on the *nd-list* of `bald`. If it is not known whether Prospero is bald or not, he appears on neither list and any query of his baldness returns `U`. Suppose the last of these is the case. If "Prospero is bald" is then input to the system, this will be a case of new information coming in that should update the model by putting Prospero on `bald`'s *d-list.* But which module is going to add Prospero (`P`) to `bald`'s *d-list*? If `P` is added to `bald`'s *d-list* by the semantic evaluation procedure for `bald`, this contradicts what Gunji has said about semantics treating the model as read-only memory. But if `bald`'s evaluation procedure does not do this update, this means that the update is somehow done by the super-interpreter after the sentence has been processed. This seems fine for the present example but does not work for example (34a): "Prospero has a car." Assume it is not known whether Prospero has a car and so evaluation of this sentence returns `U`. The super-interpreter will now want to stick one of the cars it has found that are in the model already onto the *d-list* of `have` linked to `P`. But how does it know which car of the cars in its model to do this to? I cannot puzzle any of this out from Gunji's thesis. Problems of this kind, i.e. processing utterances whose truth values are undefined in the model, confounds Gunji's projection solution too.

10.4.2 Pragmatics in Gunji's System

The basis of Gunji's system was presented in the previous subsection. To achieve pragmatic processing, Gunji has associated three procedures with each EIL lexical item: a PREC (precondition) procedure, a DEN (denotation) procedure and a POSTC (postcondition) procedure. The PREC of an item is evaluated first. It may update the model with information representing conventional implicatures and presuppositions triggered by the item.[20] Next the DEN is evaluated: this cannot update the model. It retrieves denotations (e.g. truth-values), and is basically what was described in the previous subsection. Finally, the POSTC may update the model with any generalised conversational implicatures or it may change the status of information added by previously executed PRECs. At the end of evaluating the utterance reading, the super-interpreter may also change the status of information added by PRECs and then return some response to the user. A PREC or a POSTC, when it adds information to the model, gives it a temporary label of *a-info*, meaning it is abortable. *a-info* is very much like Gazdar's p-presuppositions. Once added, a subsequent POSTC or the super-interpreter may change *a-info* into *c-info*, meaning that it is cancellable by later utterances, or may change it to *nc-info*, meaning it is not cancellable

[20] Gunji refers to the presuppositions of simple, positive utterances as conventional implicatures. I shall refer to them as presuppositions.

by later utterances, or may remove the *a-info* altogether.

The PREC of a presupposition-triggering item (such as `regret`) adds the presupposition (e.g. sentential complement of `regret`) to the model as *a-info*. In a simple, *positive* utterance, the next time this *a-info* will be looked at is at the end of utterance processing, at which point the super-interpreter may change the *a-info*'s status:

- If the *a-info* is contradictory to *nc-info* in the model, the user is notified of a contradiction. If it is contradictory to *c-info* already in the model, the *c-info* is removed (cancelled) and the *a-info* becomes *nc-info*; this is the case of cancelling a presupposition set up earlier in a text.
- If the *a-info* is repeated by *nc-info* or *c-info* in the model then the *a-info* is absorbed by the *nc-info* or *c-info* (this means it is deleted because it is already there in stronger form).
- If the *a-info* is neither contradictory nor duplicate then it is changed to *nc-info*. By changing it to *nc-info*, it will not be cancellable by information in subsequent utterances. This is what is wanted for the presuppositions of simple, positive utterances.

Negation is called a "cancellability-inducing operator". It is not at all clear how Gunji determines the scope of negation and whether it is vague (wide-scope) negation or unambiguously narrow-scope negation, particularly as definite NPs are translated as constants and so cannot fall within the scope of negation. Cancellability-inducing operators have special POSTCs. In a simple, *negative* utterance, the PREC of the presupposition-triggering item will add the presupposition as *a-info* but before the super-interpreter can get to this, the POSTC of the cancellability-inducing operator (the negation) will intervene. This POSTC will change the status of the *a-info* as follows:

- If the *a-info* is contradictory to *c-info* or *nc-info* in the model then it is aborted (deleted). This is the case of a potential presupposition being filtered out.
- If the *a-info* is repeated by *nc-info* or *c-info* in the model, then the *a-info* is absorbed by the *nc-info* or *c-info* (by deleting the *a-info* and leaving the *c-info/nc-info*).
- If the *a-info* is neither contradictory nor duplicate then its status is changed to *c-info*. This leaves the information as a presupposition but makes it cancellable by later text. This is what is wanted for the presuppositions of negative utterances.

10.4.3 Compound Utterances in Gunji

I will now consider Gunji's solution to the projection problem for conjunctions, disjunctions and conditionals. Gunji has an account which, like my own, depends heavily on the idea of left-to-right sequential evaluation. However, as he is taking a 'presuppositions as ordinary inferences' view, he does not use the idea of finding entities in a discourse model. Indeed, he does not separate context from possible worlds. He is depending on truth in the model (still in the model-theoretic sense of this word) and special definitions of the EIL operators and, imp and or (for conjunctions, conditionals and disjunctions respectively).

• Processing Conjunctions

The valuation function of the EIL operator and, v(and p q), is:[21]

```
v(and p q) is defined as {val := v(p)
                          if val then v(q)
                          else if (not val) then F
                               else if (not v(q)) then F
                                    else U}
```

Crucially q is only evaluated if p returns T or U. Furthermore, there is no question of evaluating these conjuncts in parallel or in reverse order since q is evaluated only in the context of the changes made by p's evaluation, i.e. because evaluation of p might update the model, (and p q) is not necessarily equivalent to (and q p).

A simple example is (36a) whose EIL representation is (36b):

(36) a. "*Lear is sane and he regrets he hit the fool.*"
b. (and (sane Lear) (regret Lear (hit Lear Fool)))[22]

We need to consider three cases: where (sane Lear) is T, F or U in the model to start with.

1. If (sane Lear) is T, the second conjunct is processed. (hit Lear Fool) is added as *a-info* by the PREC of regret. Assuming the super-interpreter finds that this *a-info* is neither repeated nor contradicted by existing information in the model, it can promote the status of this *a-info* to *nc-info*. This is correct.

2. If (sane Lear) is F, the second conjunct is not processed. Hence the presupposition will not get generated. This seems reasonable in this framework.

3. If (sane Lear) is U, the second conjunct is processed. (hit Lear Fool) is added as *a-info*. Assuming the super-interpreter does not find either contradictory or duplicate information that was in the model to begin with, the *a-info* will become *nc-info*.

It is the last case that is interesting. It certainly seems correct that the model should be left with (hit Lear Fool) as *nc-info* in it (this being the presupposition), but we should not leave (sane Lear) as U since this is what

[21] I have simplified this in unimportant ways.

[22] Gunji would give (and (sane Lear) (regret Lear (int (hit Lear Fool)))) where int shows that this argument of regret is intensional. I will leave this out as it does not affect the behaviour of presupposition projection.

the speaker is asserting to be true. Gunji, describing a similar example, manages to overcome this problem, but his description of the processing involved in doing so conflicts with his description elsewhere. Earlier in the thesis he says that the super-interpreter is invoked at the *end* of processing an utterance. Now he allows the super-interpreter to be invoked *in between* the conjuncts. This is inconsistent with his previous statement and is also not in any way revealed by his valuation procedure for **and**. If the latter is the way the system works, then the aforementioned problem is overcome. The first two cases are not affected (where `(sane Lear)` is T or F). In the third case, where `(sane Lear)` is U, *before* going on to the second conjunct, the super-interpreter is invoked and adds `(sane Lear)` as *nc-info* to the model. The second conjunct is processed as before, producing `(hit Lear Fool)` as *nc-info* also. This is now the correct prediction.

But this raises questions about what to do with `or` and `imp`: is processing interrupted to invoke the super-interpreter or not? Mercer says Gunji should be uniform about where the super-interpreter is invoked, i.e. if it must be invoked in the middle of a conjunction, it should also be invoked in the middle of disjunctions and conditionals. I do not necessarily agree — we can see the justification for invoking it in the middle of **and**: conjoined sentences are a bit like separate, contiguous sentences in a discourse (e.g. "*Lear is sane and he regrets he hit the fool.*" is like "*Lear is sane. He regrets he hit the fool.*") which disjunctions and conditionals are not. But we have to ignore Mercer's criticism concerning the non-uniformity of having the super-interpreter invoked in the middle of conjunctions but only at the end of conditionals and disjunctions, because invoking the super-interpreter in the middle of conditionals and disjunctions will not give correct predictions. If we can overlook Gunji's inconsistency, I think that Gunji's analysis of conjunctions is compatible with intuitions. To see the problems of his system further we must look at conditionals and disjunctions.

- **Processing Conditionals**

The valuation function for the conditional operator, `imp`, is:

```
v(imp p q) is defined as {val := v(p)
                          if val then v(q)
                          else if (not val) then T
                               else if v(q) then T
                                    else U}
```

This is very similar to the function for **and**. Crucially from a presuppositions point of view, it shares with the function for **and** the principle that q is only evaluated after p and only if p is T or U.

In the examples, I will only consider the case where the truth-value of the antecedent is U. This is the most important case and, ironically, the case Gunji does not describe. It is important because it is perhaps the normal case that the hearer does not know the truth-value of the antecedent.

(37) a "*If Lear is sane, he regrets he hit the fool.*"[23]

[23] If the super-interpreter were to be invoked in the middle of the utterance, `(sane`

b. (imp (sane Lear) (regret Lear (hit Lear Fool)))

Given that (sane Lear) is U, the consequent is processed. (hit Lear Fool) is added as *a-info* by the PREC of regret and the super-interpreter, not finding this as duplicated or contradicted, will convert the *a-info* to *nc-info*. Gunji thus correctly predicts that (hit Lear Fool) is a presupposition.

Now consider (38):

(38) a. "*If Lear hit the fool, he regrets he hit the fool.*"
b. (imp (hit Lear Fool) (regret Lear (hit Lear Fool)))

If (hit Lear Fool) is U, nothing is added to the model and the consequent is processed. (hit Lear Fool) is added as *a-info* by the consequent. The super-interpreter does not find any duplication or contradiction (because (hit Lear Fool) has not been added by the antecedent) so the *a-info* becomes *nc-info*. A presupposition is thus incorrectly predicted.

- **Processing Disjunctions**

The valuation function for the disjunction operator, or, is:

```
v(or p q) is defined as {val = v(p)
                          if val then T
                          else if (not val) then v(q)
                                else if v(q) then T
                                     else U}
```

In this case q is only evaluated after p and only then if p is F or U.

Again I will concentrate on the case (not described in Gunji) in which the first clause evaluates to U.

(39) a. "*Lear is sane or he regrets he hit the fool.*"
b. (or (sane Lear) (regret Lear (hit Lear Fool)))

(40) a. "*Lear didn't hit the fool or he regrets he hit the fool.*"
b. (or (not (hit Lear Fool)) (regret Lear (hit Lear Fool)))

In (39), the correct prediction is made. If the first disjunct, (sane Lear), is U, the second disjunct is processed: (hit Lear Fool) is added as *a-info* and then promoted by the super-interpreter to *nc-info*. This correctly predicts a presupposition.

In (40), if (hit Lear Fool) is U then so is the whole first disjunct, (not (hit Lear Fool)); before going on to the second disjunct the POSTC of not is invoked. Exactly what this is going to do is not clear. It cannot really add anything to the model because, with a disjunction, nothing has been

Lear) would be added to the model as *nc-info*. This would be quite wrong. The super-interpreter must be invoked at the end of the utterance: processing of conditionals (and disjunctions) will be different to the processing of conjunctions.

established, i.e. the speaker has not said that it is true that Lear did not hit the fool; she has said that it is possible he did and possible he did not. Assuming the POSTC of `not` does nothing, when the second disjunct is processed, `(hit Lear Fool)` will be added as *a-info* and the super-interpreter will promote this to *nc-info* giving an incorrect presupposition.

10.4.4 Final Remarks on Gunji

Given the failings of Gunji's system on even simple cases of conditionals and disjunctions, I do not need to consider cases of partial matches or cases of conflicting presuppositions in disjunctions. It should be clear that the system will make wrong predictions about these too. Failings on the data aside, Gunji's approach has many similarities with my own, in particular in its reliance on left-to-right processing. A major difference, though, is that for me presuppositions are satisfied by discourse model entities, where the discourse model is merely contextual knowledge that can be taken as mutual or familiar to both conversational participants. For Gunji, however, presuppositions are evaluated against a model of the world. So for Gunji presuppositions are inferences that are taken to be true, while for T42 they are merely demands that the context (discourse model) contains certain entities that have been talked about already. This is what causes Gunji problems in projection. He cannot handle examples such as (38a) ("*If Lear hit the fool, he regrets he hit the fool.*"), because the only way in which the presupposition will be repeated by information already in the model is if the antecedent updates the model with that information. But doing so would commit the system to believing that Lear hit the fool, and clearly it cannot so-commit itself because the conditional signals that it is only a possibility that Lear hit the fool. Gunji really requires a way of tentatively asserting the antecedent of the conditional while evaluating the consequent, and then undoing the effects of this tentative assertion at the end of the sentence. This is what I achieve using accessibility and the distinction between the ILC and the NIC.

10.5 Van der Sandt

Van der Sandt's (VdS) approach to presupposition projection was first published in Dutch in 1982, but did not appear in English until 1988 [Van der Sandt 1988]. Accepting the view of presuppositions as ordinary inferences, VdS emphasises their context-dependency. The set of presuppositions of a sentence will be those which the context requires for the utterance to be felicitous. He also stresses the importance of the left-to-right ordering of text in accounting for presuppositional phenomena: "When a sentence is asserted or entailed prior to the mention of a sentence that presupposes it, our intuition is that the resulting whole does not presuppose this sentence. However, if a sentence that is presupposed by one of the components of the text is neither asserted before, nor entailed by, a previously asserted sentence, then intuitively the whole text does presuppose this sentence." [VdS 1988, p.161].

VdS uses the notion of a context-set, which is a set of propositions taken to be true at some point in a discourse; it is the set of propositions the speaker is willing to assume at a given moment, and will include both shared background information and propositions introduced by the conversation (if they have not subsequently been challenged or retracted by the conversational participants). The presuppositions of a sentence uttered in a particular context are roughly the propositions that the sentence has to have in the context-set as a condition on interpretability of the sentence. This seems similar to my notion of presuppositions as preconditional inferences on a discourse model, but there are two differences: VdS uses propositions instead of discourse model entities, and VdS will add a presupposed proposition to a context-set as long as it is consistent to do so while I suggest that one should only be added if it is plausible to do so.

VdS's presupposition theory is thus a set of principles of "contextualisation" or "context-selection". A sentence ϕ (compound or otherwise) will presuppose a proposition χ in a context c if:

1. χ belongs to the *elementary presuppositions* of ϕ, and
2. ϕ is *acceptable* in the χ-extension of c, i.e. the context that only differs from c in that its context-set is extended by χ.

To understand this further, we need to look at VdS's theories of elementary presuppositions and of acceptability.

10.5.1 Elementary Presuppositions

VdS basically uses the usual set of presupposition triggers, e.g. definite NPs, factive verbs, aspectual verbs, and so on. However, he disputes the idea that elementary presuppositions are straightforwardly associated with specific lexical items or syntactic constructions. So, for example, while VdS wants (41b) to be an elementary presupposition of (41a), he does not want there to be such a presupposition for (41c) (otherwise (41c) presupposes what (41c) asserts) or for (41d) (otherwise (41d)'s presupposition contradicts what (41d) asserts), and he wants (41f), not (41b), to be the presupposition of (41e):

(41) a "*The chief of Buru Buru is bald.*"
b. 'The chief of Buru Buru exists.'
c. "*The chief of Buru Buru exists.*"
d. "*The chief of Buru Buru doesn't exist.*"
e. "*The chief of Buru Buru was bald.*"
f. 'There *was* a chief of Buru Buru.'

He says: "A precise specification of such a function [for computing elementary presuppositions] requires a syntactic theory that associates specific expressions with the basic vocabulary, and then determines the fate of these expressions under the mode of composition." [VdS 1988, p.195]. He later says: "A specification of the presupposition function requires a syntactic theory and falls outside of the scope of this book." [VdS 1988, p.196]. I regard VdS's desire for this kind of elementary presupposition function (which can. amongst other things, take into account verb tense as in (41e), for example) as probably unachievable in a computational system. One needs only to consider the difficulties of getting the elementary presuppositions of sentences such as (42a). Is the presupposition (42b)?

(42) a. "*The Prince of Wales became king in 1413.*"
b. 'There was a person who had the title 'Prince of Wales' until some point in 1413.'

I regard it as a success of T42 that it does not need a presupposition triggering mechanism of the complexity of the one VdS envisages. Rather, presupposition triggers depend only on the parts of an utterances that fall within "given" environments. These furnish a description which, along with salience constraints (in a fuller system), are used to retrieve an entity, and which have the potential to do this successfully even in cases where the description is inaccurate.

10.5.2 The Acceptability Relation

I will paraphrase VdS's conditions defining acceptability to avoid having to explain all his notation. Note that the definition is recursive.

A *sentence* is *acceptable* in a *context*:

1. only if the proposition it expresses is not already entailed by the context-set (else it will be redundant),

2. only if the proposition it expresses is consistent with the context-set,

3. if the sentence does not contain a coördinating construction but does contain a sentence ξ embedded under a construction such as "possibly" (he excludes verbs of propositional attitude) and ξ is not an elementary presupposition, only if ξ is acceptable in the context-set,

4. if the sentence is a conjunction or conditional, only if the first clause is acceptable in the context-set and the second clause is acceptable in the context-set augmented by the proposition expressed by the first clause,

5. if the sentence is a disjunction, only if the first disjunct is acceptable in the context-set augmented by the proposition expressed by the second disjunct, and the second disjunct is acceptable in the context-set augmented by the proposition expressed by the first disjunct.

Note that this is a definition of the acceptability of a *sentence* in a context. It has nothing to say about presuppositions at this stage.[24]

[24] Two immediate problems with this definition are that clause (1) means that it is not acceptable to repeat a sentence, and clause (2) means that it is not acceptable for someone to try to change your beliefs.

With this definition of acceptability of a sentence in a context, we can explain when an elementary presupposition will be an actual presupposition of an utterance:

> A proposition will be a presupposition of a sentence in a context if it is an elementary presupposition of the sentence; if it can be consistently added to the context-set; if it is consistent with the other elementary presuppositions; and if the assertion of the sentence is acceptable (by the above definition) in the context once the context-set is augmented with the presupposition.

At the heart of this lies the idea of determining whether a sentence uttered against a particular context-set would be acceptable if uttered against a context-set comprising the original context-set augmented with an elementary presupposition of the sentence. If it is not acceptable, this does not mean that the sentence is anomalous in any way. It simply means that the elementary presupposition is not a presupposition of the sentence in that context: the elementary presupposition is "neutralised".

10.5.3 Some Examples

Provided there is nothing untoward in the context, (43c) is a presupposition of (43a) and (43b):

(43) a. "*The Prince of Denmark smiled.*"
b. "*The Prince of Denmark didn't smile.*"
c. 'There is a Prince of Denmark.'.

For (43a) the reasoning is as follows. (43c) is an elementary presupposition of (43a), it can be consistently added to the context-set, and (43a) will be acceptable in a context-set augmented with (43c). The last of these is verified using the definition of acceptability. In fact, only conditions (1) and (2) of the definition are applicable: (43a) is not already entailed by the context-set and is not inconsistent with the context-set. Hence, (43c) *is* a presupposition of (43a). A similar argument shows that (43c) is a presupposition of (43b).

VdS gives a simple test for showing that a sentence is acceptable in a context-set augmented by a presupposition, which will enable us to avoid referring to his definition of acceptability in detail. This is to judge whether an utterance comprising the presupposition followed by the sentence is felicitous. For example, since utterance of (43c) can be felicitously followed by (43a) or (43b), as in (44a) and (44b), the presupposition survives:

(44) a. "*There is a Prince of Denmark. The Prince of Denmark smiled.*"
b. "*There is a Prince of Denmark. The Prince of Denmark didn't smile.*"

However, the elementary presupposition (43c) does not survive in (45a) because (45a) would not be acceptable in a context-set augmented with (43c), as shown in (45b) (i.e. condition (2) of the acceptability definition would be violated when assessing the second clause of the utterance in a context-set augmented by the assertion of the first clause). But (45a) does have another elementary presupposition, (45c), which does survive, as indicated by the felicity of (45d):

(45) a "*The Prince of Denmark didn't smile because there isn't a Prince of Denmark.*"
b. *"*There is a Prince of Denmark. The Prince of Denmark didn't smile because there isn't a Prince of Denmark.*"
c. 'Denmark exists.'
d. "*Denmark exists. The Prince of Denmark didn't smile because there isn't a Prince of Denmark.*"

I will now consider some of the examples of compound utterances. In the following, the elementary presupposition is that Lear hit the fool (from the factive complement):

(46) a. "*Lear hit the fool and he regrets he hit the fool.*"
b. ??"*Lear hit the fool. Lear hit the fool and he regrets he hit the fool.*"
c. "*If Lear hit the fool, he regrets he hit the fool.*"
d. ??"*Lear hit the fool. If Lear hit the fool, he regrets he hit the fool.*"
e. "*Lear didn't hit the fool or he regrets he hit the fool.*"
f. ??"*Lear hit the fool. Lear didn't hit the fool or he regrets he hit the fool.*"

Utterance of (46a) is not acceptable in a context-set augmented with the elementary presupposition, as can be seen in (46b) (also, from the definition of acceptability, adding the elementary presupposition to the context-set means that the first clause of the utterance will be entailed by the context-set and this violates condition (1) of the definition). Hence, the presupposition is correctly "neutralised". This is the case of a presupposition in one part of an utterance being asserted in another part of the utterance. Utterance of (46c) is also not acceptable in a context augmented with the elementary presupposition, as can be seen in (46d), for the same reason as for (46a). Again the presupposition is correctly "neutralised". Similarly again in (46e), the presupposition is correctly "neutralised" because utterance of the first clause in a context augmented with the elementary presupposition of the second clause is inconsistent, as can be seen in (46f).

It is important to note that none of these utterances ((46a), (46c) and (46e)) is anomalous. All that we have shown is that they are not acceptable in contexts augmented by one of their elementary presuppositions, and all this means is that the presupposition is "neutralised".

VdS's solution works on the cases of partial matches and works in part on the cases of entailed entities:

(47) a. "*If Launce owns a puppy, his dog likes him.*"
b. "*Launce owns a dog. If Launce owns a puppy, his dog likes him.*"
c. "*If Launce has a grandchild, his child is happy.*"
d. "*Launce has a child. If Launce has a grandchild, his child is happy.*"

VdS can correctly predict the ambiguity of (47a). He can obtain the reading of (47a) in which a dog is presupposed, since, as can be seen from (47b), the utterance is acceptable in a context augmented by this presupposition. Equally, he can predict the other reading. If the context-set contains the fact that Launce owns a dog which is a puppy before (47a) is uttered, then (47a) will not be acceptable in that context, and so the presupposition will be "neutralised". Thus, it would seem that VdS's theory is the only one, other than my own, which can correctly predict the ambiguity of examples such as (47a).

However, I believe VdS predicts that (47c) usually presupposes that Launce has a child (witness the acceptability of (47d)). For me, (47c) is ambiguous: it may presuppose Launce has a child, but it also need not presuppose this if it takes the child to be as hypothetical as the grandchild.[25] VdS will only get this other reading if the context-set contains something prior to utterance processing which says that Launce does not have a child, which can therefore "neutralise" the presupposition.

I will now look at VdS's predictions for cases in which the presuppositions of two different clauses conflict:

(48) a. "*Lear regrets he hit the fool or he regrets he didn't hit the fool.*"
b. "*Launce met the King of Denmark or he met the President of Denmark.*"

In (48a), the elementary presupposition that Lear hit the fool conflicts with the elementary presupposition that Lear did not hit the fool. Hence, VdS predicts that there is no presupposition for (48a). Similarly in (48b), the elementary presupposition that there is a King conflicts with the elementary presupposition that there is a President. Hence, VdS predicts no presupposition for (48b). Mercer would say, in taking a 'presuppositions as ordinary inferences' view, that (48b) should, in fact, presuppose ((There is a King of Denmark) $\vee$ (There is a President of Denmark)). Since this is not a tautology (because a country may have a queen rather than a king or a president), this disjunction is a substantive presupposition of (48b) which VdS does not find.

10.5.4 Final Remarks on Van der Sandt

VdS is very successful with the data, but there are two problems. First, his approach to elementary presuppositions seems unreasonable from a NLP

[25] However, the present *implementation* of T42 always gets the reading where the child is hypothetical; it would need mechanisms for plausible reasoning to get the other reading.

point of view, i.e. I do not believe that such an elementary presupposition mechanism could easily be built. Secondly, he does not get ambiguities in cases such as (47c) which I believe he should, and does not get a presupposition in cases of conflicting but non-exhaustive presuppositions such as (48b).

Apart from this, Van der Sandt's account has many similarities with T42's, notably the way it relies on considering the ordering of clauses in a compound utterance or text, and the way it considers the demands that utterances place on the context. These similarities are reinforced by other work by VdS. In [Van der Sandt 1987], VdS relates his theory of contextualisation to Kamp's DRT. He shows that DRT accessibility can help to determine whether a presupposition makes an extra- or intrasentential demand on the context. In relation to VdS, therefore, my main claims are that my view of presuppositions as preconditions that demand discourse model entities handles the data as well as his, and secondly is developed within a more realistic NLP framework. To some extent, if we view Weischedel's work as a computer implementation of K&P's theory, and Mercer's and Gunji's work as an implementation of Gazdar's theory, then we might view my work as an implementation of VdS's theory. (This is an 'after the event' rationalisation: there are differences between the two and both were completed independently).

::::

In this chapter I have analysed five main presupposition projection theories. None of the approaches accounts for all of the data. Karttunen & Peters have to resort to an informal "detachment" argument to get certain predictions right, and Gunji fails mostly on the same examples. Gazdar and Mercer, while doing much better on the bulk of the data, cannot adequately handle the cases of partial match which give rise to ambiguities. Van der Sandt does recognise these important ambiguous cases but there are at least two cases where his predictions may not be right.

K&P, Gazdar and Gunji also introduce a lot of technical 'apparatus', with no intuitive explanation to it. Mercer, Van der Sandt and T42 mostly avoid this. However, the advantage of the T42 account is its simplicity. This simplicity manifests itself both in T42's very simple presupposition triggering mechanism (unlike Van der Sandt's) and its simple account of presupposition "cancellation" or "suspension", which is not "cancellation" or "suspension" at all: presuppositions are always triggered and must always be satisfied, but they may be satisfied by things internal to an utterance and this gives rise to the effects of presupposition "cancellation". All of these consequences flow from the fundamental difference between my system and these others, namely that I take presuppositions as preconditional inferences which demand discourse model entities, not as ordinary inferences. The possible problems with T42's account concern T42's processing of conflicting presuppositions, and the question of the correct processing of disjunctions.

Chapter 11

Conclusions and Future Work

11.1 Conclusions

This thesis has presented an analysis of presuppositions and a method for determining them which has been computationally implemented. The work is thus a contribution both to linguistics and to natural language processing, as I make clear below.

- **Presuppositions as Preconditions: Linguistic Perspective**

This thesis has argued that presuppositions are best viewed as a special kind of inference: *preconditions.* Presuppositions are triggered by the use of certain words and constructions and must *always* be satisfied by finding a suitable entity in a discourse model, which is a representation of contextual knowledge. If they are not satisfied, then the reading of the utterance that licenses the presupposition does not come off. There is a distinction between presupposition failure and a presupposition being false. Failure occurs if the discourse model does not contain an entity to satisfy a presupposition. In such a situation, the utterance reading whose presupposition failed does not come off. But if a reading is obtained, it can then be assessed for truth in a variety of worlds (including worlds that are not consistent with the discourse model), and may then be true or false (or perhaps undefined) in those worlds.

The discourse model, against which presupposition satisfaction is attempted, contains both entities that have been introduced by preceding discourse and entities that are invoked by the physical discourse setting. The conversational participants may treat these entities as mutual, but the participants need not hold particular propositional attitudes towards these entities and their properties. The discourse model entities are merely entities that are taken to be familiar. This seems to be the right formulation for handling presuppositions. It allows one to utter sentences that presuppose the familiarity of fictional, legendary and even impossible objects. It is also a characterisation that might allow a treatment of inaccurate reference and even of lying, although these were not investigated in the thesis.

My approach to presuppositions includes specific proposals for the treatment of negation. Negation is not ambiguous and not vague. It unambiguously preserves presuppositions. Even in so-called presupposition "cancellation" uses of negative sentences, my thesis goes against the grain by continuing to demand that presuppositions be satisfied by discourse model entities. In this sense, there is no such thing as presupposition cancellation: presuppositions must always be satisfied. However, there *are* metalinguistic uses of natural language statements, and a presupposition "cancelling" utterance is one such statement. Metalinguistic statements are statements that characterise the relationship between discourse model entities and their denotations in a particular distinguished possible world. A presupposition "cancelling" utterance basically says that an entity has no denotation in a distinguished world. This does not stop the entity from being talked about, or stop the utterance reading in question from being assessed for truth in some other world where the entity does have a denotation.

I solve the "projection problem" for presuppositions in a similar way. Presuppositions again must *always* be satisfied for a reading to come off. However, they need only be satisfied against the discourse model. The order of presupposition triggers is crucial, and the way the discourse model is incremented as utterances are processed from left-to-right can affect the success of presupposition satisfaction. In particular, the antecedents of conditionals may introduce entities into the discourse model (specifically into the immediate linguistic context), and these entities may satisfy presuppositions in the consequents of the conditionals. But by being introduced in the antecedent of a conditional, the entity is a candidate referent but has the semantics of a universally quantified variable. Most projection problem data is then easily accounted for.

My solution to the projection problem is an attractively simple one. It requires no special mechanisms for "cancelling" or "suspending" presuppositions: all presuppositions must be satisfied, but they may be satisfied by entities which are 'internal' to a sentence and so place no special demands on the non-immediate context. One of the most important consequences of my solution is that I recognise that some cases are ambiguous: on one reading a presupposition is satisfied by something internal to the sentence, and on another reading it is satisfied by something external to the sentence, which places a demand on the non-immediate context.

I have not so far developed my theory fully for disjunctions. I developed an argument about them based on looking at their equivalence to conditionals. I tentatively suggested that, in the absence of the word "either", the presupposition projection properties of disjunctions are asymmetrical, i.e. the left-to-right ordering is paramount. But, where a disjunction begins with "either", the projection properties are probably symmetrical.

• Presuppositions as Preconditions: Computational Perspective

This thesis makes a computational as well as a linguistic claim, namely that its simple uniform treatment of presuppositions as preconditions is well suited to NLP. More specifically, the thesis argues that presuppositions, as conceived herein, are most easily and naturally computed in *incremental* natural language processing systems. I defined incremental systems to be systems that interleave syntactic, semantic and pragmatic processing with feedback from one module to another. Interleaving and feedback allow presupposition satisfaction to help the parser to choose a reading of an utterance.

My actual system, T42, is based on one described in [Haddock 1987a, 1987b] but there are a number of changes. First, Haddock describes only definite NP processing, but in this thesis I have made extensions to processing other parts of speech. In extending Haddock's system to allow the processing of verbs and indefinite NPs (and hence sentences and simple texts), "given" and "new" have been given new technical definitions. These definitions are such that lexical items that initiate "given" environments are those same lexical items that trigger presuppositions. This "given"/"new" perspective has proved useful in looking at a piece of data that has confounded all other formal and computational approaches to presupposition, i.e. the cases where an indefinite NP appears to have anaphoric behaviour. These are explained by showing that the indefinite NP appears within a "given" environment, i.e. within the scope of a presupposition trigger. The indefinite NP is then forced to behave in the same way as other things in "given" environments: it must find an entity in the discourse model to satisfy it. An example is "*If Dogberry came to the party, then the hostess must have been really glad that there was a policeman present.*".

The second major change I have made to Haddock's original work is that I have introduced a discourse model and made this distinct from the knowledge base. As I have already said, this is important for presupposition processing. With this distinction, I have also been able to draw an analogy between T42 and Discourse Representation Theory (DRT). I maintain that T42 has more flexibility than DRT, but is a good structure in which to implement DRT-style NLP because it not only captures the reference markers that are so important in DRT but can also represent referential ambiguity easily.

- **Limitations of the Work**

In the conclusions above, I have made various claims for T42 and the theory it embodies. In the next section, on possible future work, I indicate some of the underdeveloped parts of both the system and the theory and propose that they be investigated further. Here I wish to make explicit the more general limitations of the work reported in this thesis. So unlike the next section, this is not intended to bring out *specific* areas requiring further attention, but rather covers broader issues that I have mostly ignored and which need to be kept in mind when evaluating this work.

First, I have neglected the issue of lexical semantics. This thesis has, for ease of exposition, and not by way of making any linguistic or computational claims, mostly mapped English words to single, simple logical predicates. Revising this might have both computational and linguistic ramifications. For example, from a computational point of view, to use finer-grained constraints to represent a word's meaning might render T42's *naïve* constraint satisfaction too inefficient.

The second area I have neglected is the processing of longer stretches of discourse. T42 and the theory it embodies do not address issues that can only be investigated if one is looking at longer discourse. Again, there might be questions about the adequacy of some of the computational mechanisms, e.g. naïve constraint satisfaction, or backtracking in the parser, and there might be questions about the linguistic theory, e.g. the interaction between presupposition satisfaction and failure and degrees of salience of discourse model entities.

Despite these broad limitations, I believe that T42 provides a good framework for presupposition handling and a strong base from which to consider further work such as that described in the next section.

11.2 Future Work

I have divided the possibilities for future work into two. In the first I explore linguistic phenomena that are mentioned in the body of the thesis but which have not been given any theoretical or computational treatment in my work so far. In the second I consider the computational mechanisms described in the thesis to determine which of these would warrant further investigation and development.

- **Linguistic Phenomena that Warrant Attention**

The linguistic phenomena mentioned below have all been reviewed in the body of this report but need further consideration.

Negation needs more examination. Up to now I have only considered uses of the word "not" where it occurs as a sister to a verb phrase. There are other uses of "not" and other forms of negation, including that which is morphologically incorporated (e.g. using "un-") and that which is introduced by quantifiers such as "never" and "no one". Without more work on negation, the account given in Chapter 8 can only be regarded as partial.

Next there are the problems that arose with presupposition projection. The cases where presuppositions are embedded under verbs of saying were described but were not given a complete treatment. More study should determine the logical forms to give to both quotational and non-quotational uses of these verbs so that only appropriate inference rules may apply to them. Equally, we need to know more about how to detect whether a use of one of these verbs in a particular context is quotational or not.

For a more complete account of presupposition projection in sentences containing coördinating constructions, more examination of the data concerning accessibility is needed. Indicating accessibility was left to the user of T42. I need to consider, for example, under what circumstances modal subordination is possible.

My treatment will also remain incomplete until disjunction is researched further. In the thesis, I make a proposal that disjunctive utterances are symmetric with respect to presupposition projection if they begin with "either", but are otherwise asymmetric. If this is right, appropriate computational mechanisms for achieving this still need to be devised.

Attributive uses of definite NPs have emerged as very important. The thesis makes no attempt to consider how such uses might be detected, and looks only briefly at how their meaning might be represented: no single form seemed to suffice. However, the fact that they can be anaphoric antecedents suggests that a DRT/T42 style of representation (i.e. the use of a reference marker or discourse model entity) is needed, but we must make sure that such a representation will receive a correct model-theoretic interpretation.

This thesis presents a number of metalinguistic uses of natural language statements. Metalinguistic uses were noted for simple positive and negative natural language utterances. More work on detecting such uses and knowing quite what to do with them is required. For example, how do they change the discourse model or affect the hearer's beliefs?

The suggestions for further work given above emerge directly from noting what the thesis would require to be a more complete treatment. Beyond these, there are more general but related topics that can be investigated. For example, there is the general question of syntactic and semantic coverage. While the system handles a number of forms of noun phrases, other parts of speech need more work, e.g. aspectual verbs such as "stop", iteratives such as "again", and constructions such as it-clefts.

There is also the issue of other forms of linguistic inference, particularly implicatures. T42 already handles one phenomenon which I have suggested is a conventional implicature: the uniqueness constraint on definite NPs. This conventional implicature takes the form of a check, in this case, on the cardinality of a set of candidate referents. There may be the possibility of extending this approach to all or to a large subclass of conventional implicatures, i.e. perhaps certain other conventional implicatures should be treated as checks of some kind. For example, the occurrence of "but" in the utterance "*Helena loves Bertram but Bertram does not love Helena.*" is generally taken to license the inference 'There is a contrast between the two facts that Helena loves Bertram and Bertram does not love Helena'. Perhaps this should be something to *check*, i.e. a constraint on felicity.

As for conversational implicatures, Mercer's approach to presuppositions [Mercer 1987] might actually be more usefully used to compute some of these. In other words, they would be computed as default inferences. This is certainly Levinson's conception of generalised conversational implicatures [Levinson 1988]. But Levinson's account of pragmatic intrusion, described in Section 2.3, suggests that we need to make these inferences in an incremental system, such as T42. This leads us into the other side of the future work that could be done: the computational issues.

- **Computational Issues that Warrant Attention**

The above considered the question of linguistic coverage. Here I consider the phenomena that my work on incremental language processing for presuppositions suggests might be worthy of further investigation and experimentation.

The question of knowledge representation is always important in NLP, and T42, with its division between the discourse model and the knowledge base, raises further issues which need much more work. For example, so far I have ignored the conversational participants' propositional attitudes towards information in the discourse model, particularly their attitudes about whether an entity has a denotation or not in particular possible worlds. It is not clear how this information is best represented. An "indexing" scheme would have to be very complicated: for any entity or property of an entity it must show whether a specified conversational participant believes the entity has a denotation (or believes the property of the entity is true) in a specified possible world. Work on this would be closely related to work on general knowledge representation and belief revision.

My work so far also assumes that the discourse model contains 'salient' information, where this information has salience by virtue of having been introduced by the current discourse. By using only small carefully constructed examples in this thesis, questions about how this set of salient entities and properties is delimited and changes over time have been ignored. There are also questions about different degrees of salience *within* the discourse model and the use of more specific notions such as local and global focus. These issues have all been examined elsewhere, but their interrelationship with incremental processing and presupposition satisfaction remains open to some investigation.

I have shown that T42 needs to make certain inferences in order to account for some of the presupposition data. Its mechanisms for this are crude, but have sufficed for demonstrating presupposition processing herein. To be able to process more realistic texts, we must look at the issues of building an inference engine that can interact suitably with the rest of the language processing system. Not only is there the demanding requirement that plausible inferencing mechanisms be developed, but even the issues of doing inference in a constrained way, at the service of language processing, remain very much in need of attention.

More generally, it is not clear that T42 represents the best design of an incremental system. It consults the discourse model frequently and makes decisions as early as it can. If it cannot resolve something as early as it would like, the analysis does not come off. A more flexible system is needed. Perhaps the decision to find something unacceptable should only be made at NP and clause boundaries, or perhaps we should allow some decisions to be pended until relevant information is provided in subsequent discourse. Such broad design issues can only be resolved by further experimentation, preferably on longer texts.

::::

I have concluded by showing that there are two interrelated directions for further work. One is to develop the theory which T42 embodies and the other is to extend the mechanisms which a system such as T42 uses. The two are related because both will benefit from being rigorously tested in a computational framework.

References

Allen,J.F. and Perrault,C.R.(1980) 'Analyzing Intention in Utterances', *Artificial Intelligence, vol.15,* 143-178

Altmann,G.(1987) 'Modularity and Interaction in Sentence Processing' in Garfield,J.L. (eds.), *Modularity in Knowledge Representation and Natural-Language Understanding,* MIT Press, pp.249-257

Atlas,J.D. and Levinson,S.C.(1981) 'It-Clefts, Informativeness and Logical Form: Radical Pragmatics' in Cole,P. (eds.), *Radical Pragmatics,* Academic Press, pp.1-61

Austin,J.L.(1962) *How To Do Things With Words,* Clarendon Press

Bobrow,R.J. and Webber,B.L.(1980) 'Knowledge Representation for Syntactic/Semantic Processing', *Proceedings of the AAAI,* Stanford, pp.316-323

Boguraev,B.K.(1979) *Automatic Resolution of Linguistic Ambiguities,* Ph.D. dissertation, University of Cambridge

Boguraev,B.K. and Spärck Jones,K.(1987) *Material Concerning a Study of Cases,* Technical Report No. 118, University of Cambridge

Brown,G. and Yule,G.(1983) *Discourse Analysis,* C.U.P.

Burton-Roberts,N.(1989) 'On Horn's dilemma: presupposition and negation', *Journal of Linguistics, vol.25(1),* 95-125

Carberry,S.(1983) 'Tracking User Goals in an Information-Seeking Environment', *Proceedings of the AAAI,* Washington, D.C., pp.59-63

Carter,D.M.(1987) *Interpreting anaphors in natural language texts,* Ellis Horwood, Chichester

Cater,A.W.S.(1981) *Analysis and Inference for English,* Ph.D. dissertation, University of Cambridge

Chafe,W.L.(1976) 'Givenness, Contrastiveness, Definiteness, Subjects, Topics and Point of View' in Li,C.N. (eds.), *Subject and Topic,* Academic Press, pp.25-55

Charniak,E. et al.(1980) *Artificial Intelligence Programming,* Lawrence Erlbaum

Chastain,C.(1975) 'Reference and Context' in Gunderson,K. (eds.), *Minnesota Studies in the Philosophy of Science, Volume VII: Language, Mind, and Knowledge,* University of Minnesota Press, pp.194-269

Clark,H.H. and Marshall,C.R.(1981) 'Definite Reference and Mutual Knowledge' in Joshi,A. et al. (eds.), *Elements of Discourse Understanding,* C.U.P., pp.283-299

Cohen,P. and Perrault,R.(1979) 'Elements of a Plan-Based Theory of Speech Acts', *Cognitive Science, vol.3(3),* 177-212

Cole,P.(1978) 'On the Origins of Referential Opacity' in Cole,P. (eds.), *Syntax and Semantics, Volume 9: Pragmatics,* Academic Press, pp.1-22

Crain,S. and Steedman,M.(1985) 'On not being led up the garden path: the use of context by the psychological syntax processor' in Dowty,D.R. et al. (eds.), *Natural Language Parsing,* C.U.P., pp.320-358

Dahl,D.A.(n.d.) *Determiners, Entities and Contexts,* Mimeo

Davidson,D.(1967) 'The Logical Form of Action Sentences' in Rescher,N. (eds.), *The Logic of Decision and Action,* University of Pittsburgh Press, pp.81-120

Donnellan,K.S.(1966) 'Reference and Definite Descriptions', *Philosophical Review, vol.LXXV,* 281-304

Fawcett,B.(1986) *The Representation of Ambiguity in Opaque Constructs,* Technical Report No. CSRI-178, Computer Science Research Institute, University of Toronto

Fillmore,C.J.(1971) 'Verbs of Judging: An Exercise in Semantic Description' in Fillmore,C.J. and Langendoen,D.T. (eds.), *Studies in Linguistic Semantics,* Holt, Rinehart and Winston, pp.273-290

Fodor,J.D.(1979) 'In Defense of the Truth Value Gap' in Oh.C.-K. and Dinneen,D.A. (eds.), *Syntax and Semantics, Volume 11: Presupposition,* Academic Press, pp.199-224

Frege,G.(1975) 'On Sense and Reference' in Davidson,D. and Harman,G. (eds.), *The Logic of Grammar,* Dickenson Publishing Company (Originally published 1892, as 'Über Sinn und Bedeutung'. In *Zeitschrift für Philosophie und philosophische Kritik,* 100, 25-50), pp.116-128

Freuder,E.C.(1978) 'Synthesizing Constraint Expressions', *Communications of the ACM, vol.21(11),* 958-966

Garner,R.(1971) '"Presupposition" in Philosophy and Linguistics' in Fillmore,C.J. and Langendoen,D.T. (eds.), *Studies in Linguistic Semantics,* Holt, Rinehart and Winston, pp.22-42

Gazdar,G.(1979) *Pragmatics: Implicature, Presupposition and Logical Form,* Academic Press

Gibson,E.A.F.(1986) *A Cooperative Response Natural Language Data Base Query System,* M.Phil. dissertation, University of Cambridge

Grice,H.P.(1975) 'Logic and Conversation' in Davidson,D. and Harman,G. (eds.), *The Logic of Grammar,* Dickenson Publishing Co., pp.64-75

Grice,H.P.(1978) 'Further Notes on Logic and Conversation' in Cole,P. (eds.), *Syntax and Semantics, Volume 9: Pragmatics,* Academic Press, pp.113-127

Gunji,T.(1981) *Towards a computational theory of pragmatics — discourse presupposition and implicature.,* Ph.D. dissertation, Ohio State University

Haddock,N.J.(1987a) 'Incremental Interpretation and Combinatory Categorial Grammar', *Proceedings of the 10th IJCAI,* Milan, pp.661-663

Haddock,N.J.(1987b) 'Incremental Interpretation and Combinatory Categorial Grammar' in Haddock,N.J. et al. (eds.), *Categorial Grammar, Unification Grammar and Parsing, Volume 1, Edinburgh Working Papers in Cognitive Science,* Centre for Cognitive Science, University of Edinburgh, pp.71-84

Heim,I.R.(1982) *The Semantics of Definite and Indefinite Noun Phrases,* Ph.D. dissertation, University of Massachusetts

Hepple,M. and Morrill,G.(1989) 'Parsing and Derivational Equivalence', *Proceedings of the 4th European ACL,* Manchester, pp.10-18

Hirschberg,J.B.(1985) *A Theory of Scalar Implicature,* Ph.D. dissertation, University of Pennsylvania

Hirst,G.(1987) *Semantic Interpretation and the Resolution of Ambiguity,* C.U.P.

Hobbs,J.R.(1979) 'Coherence and Coreference', *Cognitive Science, vol.3,* 67-90

Hobbs,J.R.(1985) 'Ontological Promiscuity', *Proceedings of the 23rd ACL,* Chicago, pp.61-69

Hobbs,J.R.(1986) 'Overview of the TACITUS Project', *Journal of Computational Linguistics, vol.12(3),* 220-222

Hobbs,J.R.(1987) *Implicature and Definite Reference,* Technical report CSLI-87-99, Center for the Study of Language and Information, SRI International

Hobbs,J.R. and Martin,P.(1987) 'Local Pragmatics', *Proceedings of the 10th IJCAI,* Milan, pp.520-523

Hobbs,J.R. et al.(1988) 'Interpretation as Abduction', *Proceedings of the 26th ACL,* Buffalo, N.Y., pp.95-103

Horn,L.R.(1985) 'Metalinguistic Negation and Pragmatic Ambiguity', *Language, vol.61(1),* 121-174

Horton,D.L.(1987) *Incorporating Agents' Beliefs in a Model of Presupposition,* Technical Report CSRI-201, Computer Science Research Institute, University of Toronto

Joshi,A.K. et al.(1984) 'Preventing False Inferences', *Proceedings of the COLING-84,* Stanford, pp.134-138

Kamp.H.(1984) 'A Theory of Truth and Semantic Representation' in Groenendijk,J., Janssen,T. and Stokhof,M. (eds.), *Formal Methods in the Study of Language,* Foris Publications, Dordrecht, pp.277-322

Kaplan,S.J.(1982) 'Cooperative Responses from a Portable Natural Language System', *Artificial Intelligence, vol.19,* 165-187

Karttunen,L.(n.d.) *Presuppositional Phenomena,* Mimeo, Department of Linguistics, University of Texas, Austin

Karttunen,L.(1973) 'Presuppositions of Compound Sentences', *Linguistic Inquiry, vol.IV(2),* 169-193

Karttunen,L.(1974) 'Presupposition and Linguistic Inquiry', *Theoretical Linguistics, vol.1(1),* 181-194

Karttunen,L. and Peters,S.(1979) 'Conventional Implicature' in Oh.C.-K. and Dinneen,D.A. (eds.), *Syntax and Semantics, Volume 11: Presupposition,* Academic Press, pp.1-56

Keenan,E.L.(1971) 'Two Kinds of Presupposition in Natural Language' in Fillmore,C.J. and Langendoen,D.T. (eds.), *Studies in Linguistic Semantics,* Holt, Rinehart and Winston, pp.44-52

Kempson,R.M.(1975) *Presupposition and the Delimitation of Semantics,* C.U.P.

Kempson,R.M.(1979) 'Presupposition, Opacity, and Ambiguity' in Oh,C.-K. and Dinneen,D.A. (eds.), *Syntax and Semantics, Volume 11: Presupposition,* Academic Press, pp.283-297

Landman,F.(1981) 'A Note on the Projection Problem', *Linguistic Inquiry, vol.12(3),* 467-471

Langendoen,D.T. and Savin,H.B.(1971) 'The Projection Problem for Presuppositions' in Fillmore,C.J. and Langendoen,D.T. (eds.), *Studies in Linguistic Semantics,* Holt, Rinehart and Winston, pp.54-60

Levinson,S.C.(1983) *Pragmatics,* C.U.P.

Levinson,S.C.(1987) 'Minimization and Conversational Inference' in Verschueren,J. and Bertucelli-Papi,M. (eds.), *The Pragmatic Perspective,* John Benjamin Publishing Co., pp.61-129

Levinson,S.C.(1988) *Generalised Conversational Implicatures and the Semantics/Pragmatics Interface,* Manuscript, University of Cambridge

Lewis,D.(1979) 'Scorekeeping in a Language Game', *Journal of Philosophical Logic, vol.8,* 339-359

Mallery,J.C.(1985) 'Universality and Individuality: The Interaction of Noun Phrase Determiners in Copular Clauses', *Proceedings of the 23rd ACL,* Chicago, pp.35-41

Martin,P. et al.(1983) 'Transportability and Generality in a Natural-Language Interface System', *Proceedings of the IJCAI,* Karlsruhe, pp.573-581

McKeown,K.R.(1985) *Text generation: Using discourse strategies and focus constraints to generate natural language text,* C.U.P.

Mellish,C.S.(1985) *Computer Interpretation of Natural Language Descriptions,* Ellis Horwood

Mercer,R.E.(1987) *A Default Logic Approach to the Derivation of Natural Language Presuppositions,* Ph.D. dissertation, Technical Report 87-35, University of Vancouver

Mercer,R.E. and Rosenberg,R.S.(1984) 'Generating Corrective Answers by Computing Presuppositions of Answers, Not of Questions', *Proceedings of the Canadian Society for Computational Studies of Intelligence,* London, Ontario, pp.16-19

Ogden,R.A.(1989) *Lexical Disambiguation,* M.Phil dissertation, University of Cambridge

Pareschi,R. and Steedman,M.(1987) 'A Lazy Way to Chart-Parse with Categorial Grammar', *Proceedings of the 25th ACL,* Stanford, pp.81-88

Pereira,F.C.(1985) 'A new characterization of attachment preferences' in Dowty,D.R. et al. (eds.), *Natural Language Parsing,* C.U.P., pp.307-319

Pereira,F.C.N. and Pollack,M.E.(1988) *A Brief Overview of the Candide Project,* Technical Report No. 450, SRI International

Pollack,M.E.(1986) 'A Model of Plan Inference That Distinguishes the Beliefs of Actors and Observers', *Proceedings of the 24th ACL,* New York, pp.207-214

Pollack,M.E. and Pereira,F.C.N.(1988) 'An Integrated Framework for Semantic and Pragmatic Interpretation', *Proceedings of the 26th ACL,* New York, pp.75-86

Prince,E.F.(1981) 'Toward a Taxonomy of Given-New Information' in Cole.P. (eds.), *Radical Pragmatics,* Academic Press, pp.223-255

Pulman,S.(1987) 'The Syntax-Semantics Interface' in Whitelock,P. et al. (eds.), *Linguistic Theory and Computer Applications,* Academic Press, pp.189-224

Pulman,S.(1989) *Events and VP Modifiers,* Technical Report No. 156, University of Cambridge Computer Laboratory

Reichman,R.(1978) 'Conversational Coherency', *Cognitive Science, vol.2,* 283-327

Reichman-Adar,R.(1984) 'Extended Person-Machine Interface', *Artificial Intelligence, vol.22,* 157-218

Reiter,R.(1980) 'A logic for default reasoning', *Artificial Intelligence, vol.13,* 81-132

Rieger,C.(1976) 'An Organization of Knowledge for Problem Solving and Language Comprehension', *Artificial Intelligence, vol.7(2),* 89-127

Ritchie,G.(1980) *Computational Grammar,* Barnes and Noble

Ritchie,G.(1983) 'Semantics in Parsing' in King,M. (eds.), *Parsing Natural Language,* Academic Press, pp.199-217

Roberts,C.(1987) *Modal Subordination, Anaphora, and Distributivity,* Ph.D dissertation, University of Massachusetts

Russell,B.(1957) 'Mr. Strawson on Referring', *Mind, vol.66,* 385-389

Russell,B.(1975) 'On Denoting' in Davidson,D. and Harman,G. (eds.), *The Logic of Grammar,* Dickenson Publishing Company (Originally published 1905, *Mind,* 14, 479-493), pp.184-193

Sadock,J.M.(1978) 'On Testing for Conversational Implicature' in Cole.P. (eds.), *Syntax and Semantics, Volume 9: Pragmatics,* Academic Press, pp.281-297

Schank,R.C.(1975) *Conceptual Information Processing,* North-Holland

Schank,R.C.(1980) 'Language and Memory', *Cognitive Science, vol.4(3),* 243-284

Schank,R.C. and Rieger,C.J.(1974) 'Inference and the Computer Understanding of Natural Language', *Artificial Intelligence, vol.5,* 373-412

Schuster,E.(1988) 'Anaphoric reference to events and actions: a representation and its advantages', *Proceedings of the COLING-88,* Budapest, pp.602-607

Sells,P.(1985) *Restrictive and Non-Restrictive Modification,* Report No. CSLI-85-28, Center for the Study of Language and Information, Leland Stanford Junior University

Shanahan,M. and Southwick,R.(1989) *Search, Inference and Dependencies in Artificial Intelligence,* Ellis Horwood

Small,S.(1983) 'Parsing as Cooperative Distributed Inference: Understanding through Memory Interactions' in King,M. (eds.), *Parsing Natural Language,* Academic Press, pp.247-276

Soames,S.(1979) 'A Projection Problem for Speaker Presuppositions', *Linguistic Inquiry, vol.10(4),* 623-666

Soames,S.(1982) 'How Presuppositions are Inherited: A Solution to the Projection Problem', *Linguistic Inquiry, vol.13(3),* 483-545

Spärck Jones,K.(1988) 'Review of 'Semantic Interpretation and the Resolution of Ambiguity', by G.Hirst', *Computational Linguistics, vol.14(4),* 91-93

Stallard,D.(1987) 'The Logical Analysis of Lexical Ambiguity', *Proceedings of the 25th ACL,* Stanford, pp.179-185

Stalnaker,R.C.(1972) 'Pragmatics' in Davidson,D. and Harman,G. (eds.), *Semantics of Natural Language,* D.Reidel Publishing Company, pp.380-397

Stalnaker,R.C.(1973) 'Presuppositions', *Journal of Philosophical Logic, vol.2,* 447-457

Steedman,M.(1985) 'Dependency and Coördination in the Grammar of Dutch and English', *Language, vol.61(3),* 523-568

Steedman,M.(1987) 'Combinatory Grammars and Parasitic Gaps' in Haddock,N.J. et al. (eds.), *Categorial Grammar, Unification Grammar and Parsing, Volume 1, Edinburgh Working Papers in Cognitive Science,* Centre for Cognitive Science, University of Edinburgh, pp.30-70

Strawson,P.F.(1950) 'On Referring', *Mind, vol.59,* 320-344

Strawson,P.F.(1952) *Introduction to Logical Theory,* Methuen

Strawson,P.F.(1954) 'A Reply to Mr. Sellars', *The Philosophical Review, vol.LXIIII,* 216-231

Strawson,P.F.(1963) 'Identifying reference and truth-values', *Theoria, vol.XXIX,* 96-118

Van der Sandt,R.A.(1987) 'Presupposition and Discourse Structure', *Proceedings of the 6th Amsterdam Colloquium at the Institute for Language, Logic and Information,* Amsterdam, pp.311-330

Van der Sandt,R.A.(1988) *Context and Presupposition,* Croom Helm

Webber,B.L.(1981) 'Discourse model synthesis: preliminaries to reference' in Joshi,A. et al. (eds.), *Elements of discourse understanding,* C.U.P., pp.283-299

Weischedel,R.M.(1979) 'A New Semantic Computation While Parsing: Presupposition and Entailment' in Oh,C.-K. and Dinneen,D. (eds.), *Syntax and Semantics, 11: Presupposition,* Academic Press, pp.155-182

Weischedel,R.M. *et al*(1978) 'An Artificial Intelligence Approach to Language Instruction', *Artificial Intelligence, vol.10,* 225-240

Wilks,Y.(1975) 'A Preferential Pattern-Seeking Semantics for Natural Language Inference', *Artificial Intelligence, vol.6,* 53-74

Wilks,Y.(1978) 'Making Preferences More Active', *Artificial Intelligence, vol.11,* 197-223

Wilks,Y.(1985) 'Right Attachment and Preference Semantics', *Proceedings of the 2nd European ACL,* Geneva, pp.89-92

Wilson,D.(1975) *Presuppositions and Non-Truth-Conditional Semantics,* Academic Press

Winograd,T.(1972) *Understanding Natural Language,* Academic Press

Woods,W.A.(1980) 'Cascaded ATN Grammars', *Journal of Computational Linguistics, vol.6(1),* 1-12