Technical Report

UCAM-CL-TR-88
ISSN 1476-2986

Number 88

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

A shallow processing approach to anaphor resolution

David Maclean Carter

May 1986

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 December 1985 by the author for the degree of Doctor of Philosophy to the University of Cambridge, King's 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

The thesis describes an investigation of the feasibility of resolving anaphors in natural language texts by means of a "shallow processing" approach which exploits knowledge of syntax, semantics and local focussing as heavily as possible; it does not rely on the presence of large amounts of world or domain knowledge, which are notoriously hard to process accurately.

The ideas reported are implemented in a program called SPAR (Shallow Processing Anaphor Resolver), which resolves anaphoric and other linguistic ambiguities in simple English stories and generates sentence-by-sentence paraphrases that show what interpretations have been selected. Input to SPAR takes the form of semantic structures for single sentences constructed by Boguraev's English analyser. These structures are integrated into a network-style text representation as processing proceeds. To achieve anaphor resolution, SPAR combines and develops several existing techniques, most notably Sidner's theory of local focussing and Wilks' "preference semantics" theory of semantics and common sense inference.

Consideration of the need to resolve several anaphors in the same sentence results in Sidner's framework being modified and extended to allow focus-based processing to interact more flexibly with processing based on other types of knowledge. Wilks' treatment of common sense inference is extended to incorporate a wider range of types of inference without jeopardizing its uniformity and simplicity. Further, his primitive-based formalism for word sense meanings is developed in the interests of economy, accuracy and ease of use.

Although SPAR is geared mainly towards resolving anaphors, the design of the system allows many non-anaphoric (lexical and structural) ambiguities that cannot be resolved during sentence analysis to be resolved as a by-product of anaphor resolution.

Preface

I would like to thank:

- my supervisor, Karen Sparck Jones, for her invaluable help and guidance over the last four years, and especially for the large amounts of time, energy and attention to detail she has devoted to reading and commenting on various versions of this thesis.

- my colleagues in the Computer Laboratory and elsewhere, for technical, theoretical and moral support. In particular: Bran Boguraev, especially for patiently answering the endless questions resulting from my early attempts to use his analyser; John Tait, for allowing me to cannibalise his remarkably user-friendly generator; John Carroll, whose enhancements to the Cambridge Lisp environment made programming so much more fun; systems gurus Graham Titmus and Piete Brooks; Richard Barber, whose support and enthusiasm made a unique contribution to the atmosphere in our office, as did his towel and swimming trunks; and Hiyan Alshawi, Ted Briscoe, Megumi Kameyama, Steve Pulman, Graham Russell, Candy Sidner and Yorick Wilks, for their encouragement and for shedding light on matters which otherwise would have remained obscure to me.

- the authors of the stories in Appendix B: Richard Barber, Ted Briscoe, Joan Carter, Gill Hooker, Antonia Lovelace, Peter Seuren, Karen Sparck Jones and others.

- the British Library, and King's College, Cambridge, for financial support (and Ken Moody, for having his brain switched on when mine wasn't).

- Douglas Holt, Priest-in-Charge at St Barnabas Church, Cambridge, for turning fragmentation into integration, in academic matters and in far more besides.

- last but certainly not least, my wife Kathy, for putting up with all that the spouse of a PhD student has to put up with. My gratitude to her is in no way lessened by the fact that I had to put up with exactly the same from her at the same time.

::::

This dissertation is the result of my own work and is not the outcome of any work done in collaboration.

I hereby state that this dissertation is not substantially the same as any I have submitted for a degree, diploma or other qualification at any other university. Further, no part of this dissertation has already been or is being concurrently submitted for any such degree, diploma or other qualification.

Contents

1. Introduction 1
    1.1 The aims of the work 1
    1.2 What is original about the work 3
    1.3 Methodology 4
    1.4 The selection of starting points for SPAR 6
        1.4.1 Boguraev's analyser 6
        1.4.2 Wilks' theory of common sense inference 7
        1.4.3 Sidner's theory of local focussing 8
    1.5 System implementation 10
    1.6 An example 12
    1.7 Thesis organisation 13

2. Anaphora in English 15
    2.1 Some basic terms and assumptions 16
    2.2 Anaphoric forms in English 19
        2.2.1 Cospecification 20
        2.2.2 Substitution and ellipsis 21
        2.2.3 Lexical cohesion 22
        2.2.4 Cohesion and semantic representation 24
    2.3 How anaphors relate to their antecedents 25
        2.3.1 An overview of work on the semantics of definite anaphora 25
        2.3.2 Discourse Representation Theory 28
        2.3.3 Webber's formalism 30
    2.4 Configurational constraints on coreference 32

3. Preference Semantics 36
    3.1 Semantic primitives and formulas 37
    3.2 Fragmentation and template matching 38
    3.3 Boguraev's analyser 40
    3.4 Unimplemented extensions to the PS framework 42

4. Meaning Representation in SPAR 43
    4.1 Requirements 43
        4.1.1 Design Methodology 43
        4.1.2 The need for a text model 44
        4.1.3 Overview of SPAR's meaning representation 46
    4.2 The representation of word senses 48
        4.2.1 Word senses in formulas 49
        4.2.2 Making formulas more explicit 51
        4.2.3 Constructing the word sense network 52
    4.3 The text model network 55
        4.3.1 Overview of the text model 56
        4.3.2 The structure matcher 59
        4.3.3 Construction of current fragments 61
    4.4 Summary and evaluation 63

5. Focussing and discourse structure 66
    5.1 Focus, foci and focussing 66
    5.2 Global focus and discourse structure 68
        5.2.1 The use of task structure: Grosz's theory 69
        5.2.2 Coherence Relations 70
        5.2.3 Cumulative treatments of focus 71
    5.3 Local focus and sentence structure 72
        5.3.1 Focus and topic in sentences 72

5.3.2 Rule-based approaches to local focus 73
5.4 Sidner's theory of local focussing 74
5.4.1 Sidner's processing framework 75
5.4.2 Interpreting full definite noun phrases 77
5.4.3 Interpreting definite pronouns 78
5.4.4 An alternative theory 81
5.4.5 An evaluation of Sidner's theory 82

6. Coordinated anaphor resolution in SPAR 85
6.1 Anaphoric, word-sense and structural ambiguity 86
6.1.1 Ambiguity resolution in textual context 86
6.1.2 SPAR's processing strategy 88
6.1.3 An example 90
6.2 The anaphor resolution rules 90
6.2.1 Rules for full noun phrases 91
6.2.2 Rules for pronouns 93
6.2.3 Assessing candidate antecedents 95
6.3 Intrasentential candidate antecedents 97
6.3.1 When an antecedent is important 98
6.3.2 Two ways to extend Sidner's algorithms 100
6.3.3 Augmenting the focus registers 101
6.4 Interactions between anaphors 103
6.4.1 Applying syntactic constraints 104
6.4.2 Incorporating the results of common sense inference 105
6.4.3 Some further collective criteria 107
6.5 Summary and evaluation 110

7. Inference and world knowledge 112
7.1 Two early story understanders 114
7.2 Specialised and heterogenous approaches to story understanding 116
7.3 Uniform approaches to story understanding 119
7.4 Inference and preference semantics 121
7.4.1 Wilks' theory of common sense inference 121
7.4.2 An evaluation of Wilks' theory 124
7.4.3 Hobbs' theory of inference 126
7.5 Anaphor resolution in a fully-specified domain 128
7.6 Summary 129

8. Inference in SPAR 130
8.1 Shallow processing, world knowledge and inference 131
8.2 The basic CSI mechanism 133
8.2.1 Extraction and CSI 133
8.2.2 Completing chains 136
8.2.3 Single-tendril chains 137
8.2.4 Negative chains 138
8.3 Constraining inference 141
8.3.1 A taxonomy of inference types 142
8.3.2 Using focussing to constrain inference 146
8.4 Extending the basic mechanism 149
8.4.1 Object-oriented inference 149
8.4.2 Goal-oriented inference 152
8.4.3 Weak inference rules 154
8.5 Summary and evaluation 155

9. Paraphrase generation in SPAR 158
9.1 What should a paraphrase achieve? 159
9.1.1 Boguraev's view of paraphrase 160

9.1.2 Applying Boguraev's argument to SPAR 160
9.2 The general organisation of the generator 163
9.2.1 Some previous work on generation 163
9.2.2 The components of SPAR's generator 167
9.3 Some guidelines for generating suitable paraphrases 168
9.3.1 The strategic component: avoiding anaphoric ambiguity 169
9.3.2 The tactical component: avoiding lexical and structural ambiguity 173
9.4 The detailed organisation of the generator 175
9.4.1 A "need-driven" control structure 175
9.4.2 A detailed example 177
9.5 Summary and evaluation 179

10. Conclusions 182
10.1 System review 182
10.2 Assessment of the system components 184
10.3 Shallow processing for anaphor resolution 187
10.4 Directions for further work 189
10.4.1 Intensional contexts and quantifier scope 189
10.4.2 Recovering from errors 191
10.4.3 Extending SPAR's linguistic coverage 193

Appendices
Appendix A: Short example texts and paraphrases 194
Appendix B: Longer example texts and paraphrases 201
Appendix C: The inference component in action 215
Appendix D: SPAR's CSI rules 220

References 223

Index of terms 230

1. Introduction

"Gratuitous mystification begins from the moment that we start to peer around for the beings named by our pronouns."

Gilbert Ryle, "The Concept of Mind", 1949

This thesis describes SPAR (Shallow Processing Anaphor Resolver), a natural language processing program which resolves linguistic ambiguities in non-specialised texts such as stories by taking account of the effects of both sentential and textual context. In SPAR, attention is concentrated on resolving anaphoric ambiguity. An anaphor is, roughly speaking, an abbreviated linguistic form whose full meaning can only be recovered by reference to the context; thus in the following sentence pair, both of the underlined phrases are anaphors.

(1-1) John bought a new car. He drove the car away.

SPAR exploits previous work by Boguraev, Wilks and Sidner, but goes significantly beyond this to achieve an integrated interpretive system, whose performance is tested by paraphrase.

This introductory chapter describes the aims of the work reported, what is original about it, and the methodology adopted in pursuing it. The theories on which SPAR is based are then summarised briefly. This summary is followed by an overview of the system and an example of its performance. Later chapters go into more detail on all these topics.

1.1 The aims of the work

SPAR uses as a front end Boguraev's [1979] English analyser, which applies syntactic and semantic knowledge to isolated sentences in order to resolve word sense (lexical) and structural ambiguity. For each sentence in a story in turn, SPAR takes as input the one or more semantic structures produced by Boguraev's analyser, selects one of the structures as correct if there is more than one, and incorporates it into its model of the story, in the process resolving any anaphoric ambiguities. It then generates a paraphrase of the sentence showing what resolutions have been made; sentence-by-sentence paraphrase allows the progress of resolution at each stage in a story to be evaluated.

At the level of system task, the aim of SPAR is the same as that of Boguraev's program: paraphrase. The two programs have the same general philosophy, one which attaches central importance to resolving ambiguity, and complement each other in attacking different aspects of the interpretation problem.

A system that resolves anaphors in stories must be able not only to exploit linguistic (e.g. syntactic and semantic) knowledge, but also to make common sense inferences (CSIs) about likely courses of events. However, unrestricted CSI is a very complex process requiring vast and varied quantities of knowledge about the world, and the development of a system capable of flexible, robust and powerful CSI is still, despite much effort, beyond the state of the art.

However, it seems plausible that a considerately-written text (i.e. one that is perspicuous and unambiguous, in accordance with Grice's [1975] maxim of manner) will normally be constructed in such a way that constraints on interpretation derived from different kinds of knowledge will tend not to conflict but rather to confirm one another and work together to guide the reader towards correct interpretations. Moreover, because language has some degree of redundancy, the same information may often be contained in more than one constraint or prediction. For example, when resolving a pronoun, we might expect that the most focussed possible referent will usually also be the one that CSI suggests is most plausible.

The second, and main theoretical, aim of this work is therefore to investigate the feasibility of a shallow processing approach according to which SPAR attempts to carry out as much anaphor resolution as possible using only linguistic knowledge, resorting to CSI only when necessary. CSI, when it is invoked, has access only to very limited quantities of general world knowledge. Under these conditions, CSI will quite often return no results, or even wrong results; however, such behaviour (we will see) need not lead to disaster.

As well as exploiting world knowledge, an anaphor resolving system must, as will be argued later, be able to maintain and use a reliable record of what characters, objects and events are more or less prominent or focussed as the story progresses. Focussed entities are more likely to serve as the referents of anaphors; but it is easy to show that a simplistic focussing mechanism based purely on recency of mention cannot reliably predict likely referents.

Accordingly, the third aim of this work, which is instrumental to the second aim of investigating the feasibility of anaphor resolution by shallow processing, is to adapt and develop existing partial solutions to the problems of focussing and CSI. The approaches chosen, for reasons to be explained in section 1.4, are those of Sidner [1979,1981,1983] for focussing, and Wilks [1975a,1975b,1976,1977a,1977b] for CSI. Wilks' formalism for word-sense semantics, in combination with Alshawi's [1983] work on representation, also plays an important part in SPAR. The process of constructing SPAR according to Wilks' and Sidner's theories brought to light a number of ways in which they should be (and were) modified and developed.

The fourth aim, whose attainment turned out to be crucially important to the performance of the system as a whole, is to find a way of coordinating these treatments of CSI and focussing (and, less crucially, the basic syntactic and semantic components of the system) so as to achieve the maximum accuracy for the minimum amount and complexity of processing. Since CSI is more complex (and, in SPAR, less reliable) than focussing, a major aspect of the coordination task is to allow focussing and other linguistic processing to dominate as much as possible, invoking CSI only when it is really essential. By doing so, the inevitable weaknesses in CSI are largely hidden or avoided by the other parts of the system, thus giving the shallow processing approach the best chance of success.

1.2 What is original about the work

SPAR represents original work in three major respects. It constitutes:

(1) the first thorough test of Wilks' inference ideas; in Wilks' implemented system, only a handful of the inference rules on which the theory is based were encoded, and later modifications to the theory were not implemented at all. I will argue that while Wilks' ideas are basically sound, his framework should be (and, in SPAR, is) extended to incorporate a wider range of inference rules than he envisaged. These extensions make the inference engine more powerful without changing its basic character.

(2) a fairly comprehensive implementation, with alterations and significant extensions, of Sidner's anaphor resolution algorithms. The algorithms are extended to deal with two phenomena which Sidner's work did not cover: first, the possibility of referents arising from the same sentence as the anaphor as well as from earlier sentences; and second, the extra problems that arise when several anaphors in a sentence impose constraints or preferences on each other's referents. These two extensions are necessary for a Sidner-derived system to have a realistic coverage of anaphoric phenomena. They result in a change in the kind of CSI that tends to be required and an increase in the complexity of the interactions between focussing and CSI.

(3) a test of the major theoretical claim of this thesis, which I will call the shallow processing hypothesis: roughly, that the shallow processing approach, of relying heavily on linguistic knowledge and strictly limiting the extent and use of world knowledge, can usually achieve accurate results. The particular form of this general hypothesis which I will attempt to substantiate is as follows:

> A story processing system which exploits linguistic knowledge, particularly knowledge about focussing, as heavily as possible, and has access only to limited quantities of world knowledge, which it invokes only when absolutely necessary, can usually choose an appropriate antecedent for an anaphor even in cases where the inference mechanism by itself cannot do so.

There are several points to note about this hypothesis.

Firstly, the hypothesis as stated is about anaphor resolution in a story processor; if it is true in this restricted case, it may well be true for other parts of the interpretation task and other types of text, but that is beyond the scope of this thesis.

Secondly, the boundary between linguistic knowledge and world knowledge is not easy to define; there is probably no principled way of deciding whether a given piece of information is part of the meaning of a word (linguistic knowledge) or is just a fact about the thing the word describes (world knowledge).[1] In this thesis, the following pragmatic view will be taken: the meaning of a lexical item consists of the handful of pieces of information that are in some sense essential to a basic understanding of that item. No ***a priori*** restriction is placed on the ***kinds*** of fact that can count as linguistic knowledge; rather, the restriction is a quantitative one. Thus whereas we might define the word "restaurant" linguistically as "a place where people eat", we would not define it as "a place where people enter, sit down, look at a menu, give a waiter an order...". Such a quantitative characterization of linguistic knowledge reflects the fact that what makes world knowledge (however defined) difficult to process is at least as much its quantity as its variety.

Thirdly, the term "***limited quantities*** of world knowledge" in the shallow processing hypothesis means that the amount of world knowledge in the system should not greatly exceed the amount of linguistic knowledge on any intuitive measure of "amount". This point, and its implications for the kind of world knowledge that is appropriate, are argued in chapter 8.

Finally, the hypothesis is one of degree; the claim is not that any system seriously deficient in CSI can ***always*** make correct decisions, but that serious deficiencies in CSI need not lead to correspondingly serious deficiencies in overall performance.

In relation to this last point, it is reasonable to ask what the consequences will be of the errors that a shallow processing system does make, and how they can, if necessary, be detected and corrected. These issues will be discussed in detail in chapter 10; they involve such factors as whether user intervention is a possibility, and whether the errors, if uncorrected, are likely to be cumulative.

1.3 Methodology

The work reported in this thesis comes into the category of engineering rather than cognitive science or linguistics. Its goal is purely that of performance: to produce a natural language processing system that resolves ambiguities correctly and generates correspondingly correct paraphrases. No claim is made that the way SPAR does this bears any relation to the way that people would go about the same task.

Similarly, although this work exploits some of the insights of theoretical linguistics, no claims are made for any contribution to non-computational linguistic theory. In particular, no strong position is taken concerning the

---

[1] Of course, aspects of the meaning of a word are also facts about what it describes, and so strictly speaking linguistic knowledge (at least about word meanings) is a subset of world knowledge. However, I will use the term "world knowledge" to refer only to knowledge that is not deemed to be linguistic.

nature of ambiguity; it is merely assumed that a natural language processor faces, and must resolve, ambiguities of at least the lexical, structural and anaphoric types. A theoretically motivated characterisation of ambiguity might lead to a different way of distinguishing particular word senses, structures and anaphoric behaviours, but I believe this would not materially affect the nature of the problems SPAR and similar systems have to face.

Similar remarks apply to the question of paraphrase. I present no rules for deciding whether two sentences or texts "mean the same thing"; for the purpose of this project, a paraphrase fulfils its role to the extent that it shows that SPAR has assigned to its input one interpretation rather than another, and it is intended to be assessed using ordinary linguistic intuitions rather than any formal decision procedure.

Unlike tasks such as translation, summarising, or paraphrasing from one sublanguage to another (e.g. Granger [1983]), SPAR's type of paraphrase is probably only of limited practical value, but it has the methodological value of providing evidence of the system's interpretive abilities to a reader who has no knowledge of its internal workings or representations. These interpretive abilities are ones that will be needed by many practically useful systems (especially translation systems) as well.

The choice of simple stories as the texts to be processed can be justified along similar lines. Their relatively straightforward content and linguistic form mean that a realistic range of phenomena can be tackled at a level of complexity that presents genuine, but usually not insuperable, problems. Of course, the processing required to resolve ambiguities in real texts (even real children's stories) would be far more complex than that performed by SPAR, but it is reasonable to suppose that the differences in complexity would be largely quantitative rather than qualitative: that is, that such processing could be done by a system in which components for focussing, CSI and syntactic and semantic judgment could be distinguished, as they are in SPAR. At least some of those components would be far more powerful than those of SPAR, and there would almost certainly be other components not corresponding to anything in SPAR. However, I believe that all the problems SPAR faces would also be faced in magnified form by a more ambitious system, and are therefore worth solving; and that the techniques developed here would be a useful basis for those needed in such a system.

The stories SPAR processes are assumed to be chronologically ordered: that is, it is assumed that events are described in the order in which they occur. In chapter 8 it will be argued that if this simplification were removed, the solutions developed to the inference problem would need to be augmented (in ways that will be outlined), but would not be invalidated.

Although the system was designed to process non-specialised texts about sequences of events, it turned out also to be able to deal with some quite different texts which were originally written to test other language processing systems. Two such texts are shown in appendix B (texts B21 and B22). This kind of flexibility is one of the spinoffs one might expect from a shallow processing approach which does not depend on detailed knowledge about any text type or domain.

Appendix B in fact contains SPAR's paraphrases of over twenty stories written by people without a detailed knowledge of the system's workings. The form of the stories was constrained only by the linguistic coverage of Boguraev's analyser and by the requirement that they describe a chronological sequence of events and states. In this way, the danger of "testing" the system only with phenomena it can already handle is largely avoided.

1.4 The selection of starting points for SPAR

A full survey of work in artificial intelligence (AI) and linguistics relevant to the problems SPAR addresses is presented in chapters 2, 3, 5 and 7. In this section, I explain briefly why Boguraev's analyser and Wilks' and Sidner's theories were chosen as the bases upon which SPAR was developed.

1.4.1 Boguraev's analyser

In order to concentrate fully on the problems of anaphor resolution, it makes sense to use an existing sentence analyser if a suitable one is available, since the two processes of assigning a semantic structure to a sentence and incorporating that sentence into a representation of the larger text (which involves anaphor resolution) can reasonably be distinguished. This is not to say that there is a clear-cut theoretical distinction between these two processes, nor that the interactions between them will necessarily be straightforward (although in SPAR they are); the point is merely that many aspects of the overall task of sentence interpretation fall naturally into one of the two classes, and from a computational point of view, modularity is usually desirable, especially if one of the modules already exists.

An important consideration in deciding to use Boguraev's analyser was the practical one that it was available and its author was on the spot to provide advice and maintenance. However, it is suitable for a number of theoretical reasons too.

Firstly, it is a general-purpose analyser in that its grammar and semantics are not biased in favour of any subject area or task to the exclusion of others. It produces case-labelled dependency structures that express the meanings of sentences in a neutral, application-independent way. These structures constitute appropriate input for a system which further processes stories written in everyday language about everyday events.

Secondly, the semantic structure assigned to a sentence is relatively shallow. It mirrors syntactic structure to a certain degree (mainly at clause and noun-phrase level), and no attempt is made to decompose or otherwise transform word meanings to produce deeper representations that might, for example, show explicitly the near-equivalence of "John gave Mary a book" and "Mary received a book from John". This shallowness of representation means that the semantic structures can be deepened or transformed as needed (if at all), which fits in well with SPAR's shallow processing methodology. It also makes dictionary entries shorter and simpler, and enables a more straightforward (although possibly less flexible) generator to be written than would otherwise be the case.

Lastly, Boguraev's semantics are derived from those of Wilks, and in fact many of the above remarks about the structures delivered by his analyser are equally true of those delivered by the parsing component of Wilks' system. A uniform semantics is clearly a desirable feature for SPAR to have, and the fact that Boguraev and Wilks share roughly the same semantic theory (preference semantics) makes this considerably easier to achieve.

1.4.2 Wilks' theory of common sense inference

In Wilks' implemented system, inference rules were invoked whenever a pronoun could not be resolved by semantic pattern matching alone. Each rule expressed a (usually causal) relationship in terms of two linked semantic patterns exploiting his general semantic primitives. The rules were used to construct chains connecting the representation of a sentence containing problem pronouns with those of sentences containing candidate referents. A completed chain would bind one or more pronouns. Inference rules were of two types: analytic <u>extraction</u> rules and non-analytic <u>common sense</u> inference rules.[2] The second type was only invoked if the first failed to bind all remaining pronouns. No attempt was made to find chains longer than a preset maximum; thus inference was guaranteed to terminate, albeit perhaps with incomplete results, within a certain time.

According to Wilks [1975b], the essential distinguishing features of his approach to inference are "(1) the inferential use of partial information; that is, information weaker than that in dictionaries and analytic (logically true) rules ... (2) the preferring of one representation or inferential chain to another." Wilks goes on to add (p55): "The common sense rules of inference used in this system are not deductive consequences about the world, but correspond to likely courses of events which, if and only if they match onto the available explicit and implicit information in the text, may be said to apply, and by applying may enable us to identify mentioned entities and so resolve problems of reference."

---

[2]The term "inference", in the sense of deriving new information from existing information or making implicit information explicit, could in principle be used to describe virtually all aspects of natural language processing. However, in this thesis, as in Wilks' descriptions of his system, it will be used in the narrower sense of finding the connections between propositions in the text. At least where story processing is concerned, such inference will in practice almost always involve the use of non-analytic, "common sense" knowledge of the world, even when some analytic knowledge is used as well. The terms "inference" and "common sense inference" (CSI) will therefore be used interchangeably when referring to the *process* of making inferences. This should not obscure the fact that in Wilks' framework (and also in SPAR), "inference rules" are of two disjoint types: extraction rules and CSI rules.

The use of "information weaker than that in dictionaries and analytic (logically true) rules" is widespread in systems which process texts about everyday, non-specialised subjects. It is Wilks' use of this information in a partial form, and also his principle of preference, which make his theory attractive for use in a "shallow processing" system like SPAR. In Wilks' framework, inference rules are not used to impose a complete organisation on the text, but merely to provide an explanation of a part of the text, and then only when a pronoun cannot be resolved by other means. Partial explanations, one hopes, are considerably easier to provide than full ones.

Wilks' principle of preference further reduces the expected complexity of inference because it states that the shortest chain of inferences (i.e. the least complex explanation) leading to the resolution of the pronoun in question should be accepted. This removes the need for any separate evaluation of the plausibility of a chain on intrinsic grounds.

The reduced complexity which is to be expected of an inference engine designed along Wilks' lines is, as should become apparent in later chapters, likely to lead to comparatively robust and flexible performance for a given amount of knowledge. This is especially desirable under a shallow processing methodology, where the resources available for CSI are limited. On these theoretical grounds alone, Wilks' ideas seem highly suitable for incorporation into SPAR, but in fact there are at least two other reasons for doing so. The first is the already-mentioned common ground shared by the semantic formalisms of Wilks and Boguraev, which should lessen the problems of inconsistency between the various strands in SPAR. The second is that Wilks' approach to inference has not been very thoroughly exploited in implemented systems. In Wilks' own system, only a few CSI rules were implemented, so the claim that the shortest inference chain would always be correct could not be properly tested; and what subsequent development of preference semantics has occurred has tended to be in the area of sentence analysis rather than inference. A serious evaluation of preference semantics in the domain of inference is therefore overdue.

1.4.3 Sidner's theory of local focussing

In Wilks' system, anaphors were resolved purely by applying semantics and rule-based inference; only if these means were inadequate was some notion of focusing brought into play. However, for a text of more than two or three sentences, some non-trivial focusing mechanism is needed for delimiting *in advance* the set of possible referents of an anaphor and imposing some structure or priority order on this set. Such a mechanism will interact with the various processes (including CSI) which assess the plausibility of possible referents. It will do so by suggesting possibilities and/or by providing a priority criterion which operates when several suggested possibilities are assessed as equally plausible.

An essential requirement of a referent-suggesting mechanism for SPAR is that it should be able to operate using only the kind of knowledge that, under the shallow processing methodology, is likely to be available. This rules out approaches such as that of Grosz [1977], in which the focus of the discourse is recognised and maintained by using detailed knowledge of the structure of the activity being discussed, to which the structure of the discussion is closely related.

In Sidner's theory, however, focus-based rules are applied to each anaphor in a sentence. The rules suggest candidate referents, normally one at a time, according to the contents of a set of focus registers which have been set during processing of earlier sentences. Suggestions are assessed by an inference component (which Sidner assumed to exist but did not specify) which uses semantic and common sense knowledge, and the first plausible suggestion is accepted. The focus registers are then updated on the basis of the resolutions made.

Thus Sidner's framework, unlike Grosz's, places no demands on the system's knowledge other than those already acknowledged for processes such as semantic judgment and CSI. In any case, in Sidner's words, "the complexity of the inferencing is constrained to asking for confirmation of the sentence predication, thereby eliminating combinatorial search for antecedents and non-terminating inferencing." (Sidner [1979],p75). This confirmation consists of "proving that the sentence with the pronoun replaced by the co-specified noun phrase is consistent with other knowledge." (p150). In other words, as long as the suggested candidate does not give rise to a contradiction, it is accepted.

Sometimes, however, the rules suggest two or more candidates at once. When this happens, a different mode of inference is invoked to judge which (if any) is most plausible. Although Sidner viewed this type of judgment as "special", it is in practice required quite often.

As mentioned briefly above, Sidner's theory is incomplete in two important respects. Firstly, the PI rules do not specify how or when candidates from the same sentence as the pronoun should be considered. Secondly, no attention is given to any possible interaction between the applications of the rules to different anaphors in a sentence. Both problems need to be tackled in a practical system, and in fact are largely solved in SPAR.

A further limitation of the theory, which Sidner acknowledged, is that it is a theory of local, not global, focus, so that it breaks down when the global structure of a discourse becomes important. However, the stories processed by SPAR consist of a linear sequence of events, and are not usually long enough for global structure to be a problem.

Like Wilks' theory of CSI, Sidner's algorithms have not been much exploited in implemented programs; every implementation to date seems to have been partial and/or in a restricted domain. SPAR thus represents the first reasonably full test (and, we will see, development) of the algorithms for non-specialised texts.

1.5 System implementation

SPAR consists of about 24,000 lines of Cambridge LISP, and runs on a GEC Series 63 in an image of about 2.5 megabytes. A story is processed roughly as follows (see also figure 1.1). Underlined terms introduced here will be defined more fully in later chapters.

First, Boguraev's analyser analyses each sentence in the story in isolation, constructing a dependency structure for each one.[3] The semantic formulas for the word senses occurring in the dependency structures are then converted into a form based on Alshawi's [1983] memory representation formalism, and combined as appropriate into a single net (the word sense network).

Next, SPAR itself is invoked. After the word sense network is loaded, the representation of each sentence in the story is processed in turn, and eventually integrated into the context provided by earlier sentences. The representation of the current sentence (after suitable transformation) and the representation of the story context together make up a text model network that is parallel to, but distinct from, the word sense network.

The stages of processing a sentence are essentially as follows.

> (1) The hierarchical dependency structure is matched with parts of the word sense network in order to bring out implicit information, and is converted into the current fragment of the text model network by a fragment constructor that makes heavy use of word sense information. One by-product of this process is the use of case-role preferences (as contained in the word sense network) to alter some network nodes representing pronouns so as to restrict their possible referents. For example, the structure for "it" in the sentence "John drank it" might be constrained so that it could only be assigned a liquid referent.

Next, a number of knowledge sources are invoked to predict or further constrain the interpretations of anaphors. The resulting predictions are evaluated by an arbitrator as soon as they are made. They arise as follows:

> (2) A set of anaphor resolution rules, based on Sidner's, is applied to each anaphor in the sentence independently. A semantic matcher is applied to assess each suggested referent; the first acceptable one is returned. If several referents are suggested at once, and more than one is semantically acceptable, CSI is not invoked immediately; instead, all the acceptable referents are returned. The reference of such an anaphor is thus constrained but not finally decided.

---

[3]In fact the analyser will produce several readings for a sentence if it cannot resolve all the lexical and structural ambiguities. This complication will be ignored in the present brief overview.

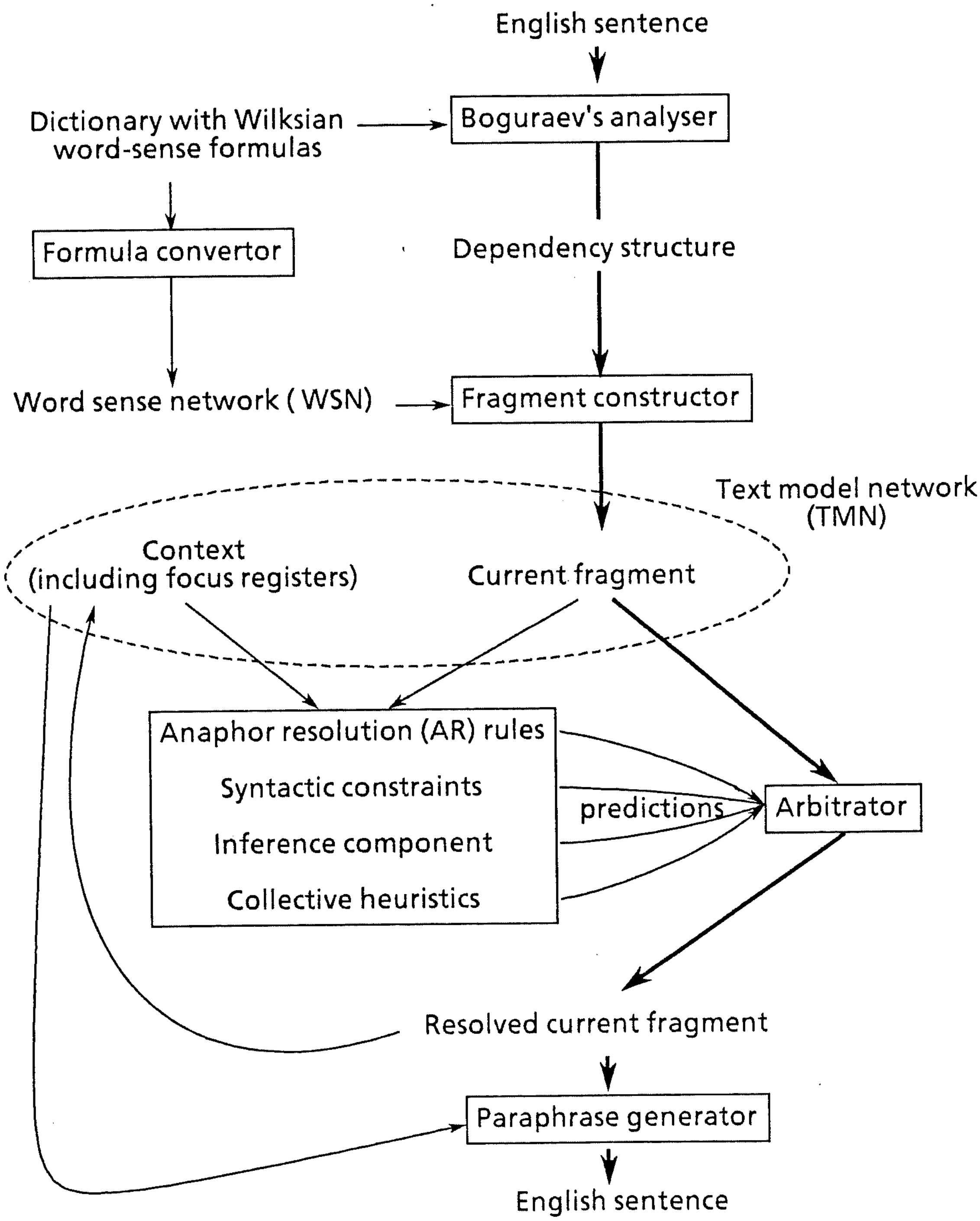


Figure 1.1: Overview of processing in SPAR.
(Heavier lines show main flow of control)

(3) A syntactic component, which makes use of approximate versions of Reinhart's [1983] c-command rule and other syntactic rules affecting coreference, is invoked to ensure consistency between the possibilities returned by stage (2). This often results in some possibilities being discarded.

If any pronouns are still unresolved, the fourth stage of processing takes place:

(4) A Wilksian CSI component is invoked in order to choose between remaining candidate antecedents for unresolved pronouns. CSI returns zero or more chains of inferences which bind pronouns to antecedents. The arbitrator accepts these chains only if they are consistent with the predictions of stages (2) and (3).

If CSI is insufficient to resolve the pronouns in question, then:

(5) an ordered set of linguistic collective heuristics, described in chapter 6, are applied until the arbitrator decides that all anaphors are resolved. Although less reliable than the earlier stages, they are normally successful on the occasions when they are required.

The current fragment is then incorporated into the "context" part of the text model network using the results of anaphor resolution. After incorporation it is handed to the last component of the system:

(6) an English generator, partly adapted from one written by Tait (Tait and Sparck Jones [1983]) to produce sentences intended to make ambiguity resolutions, especially anaphoric ones, as clear as possible.

The dependency structure for the next sentence in the story is then processed.

1.6 An example

To give some indication of SPAR's capabilities, its output for one of the stories it processes is shown below. The first sentence of each pair is the input, as handed to Boguraev's analyser; the second of each pair is the paraphrase finally produced by SPAR after all ambiguities have been resolved.

```
[1]. JOHN PROMISED BILL THAT HE WOULD MEND HIS CAR.
     JOHN PROMISED BILL THAT JOHN WOULD REPAIR BILL'S CAR.

[2]. HE TOOK IT TO HIS FRIEND'S GARAGE.
     TO JOHN'S FRIEND'S GARAGE, JOHN CONVEYED THE CAR.

[3]. HE TRIED TO PERSUADE HIS FRIEND THAT HE SHOULD LEND HIM SOME TOOLS.
     JOHN ATTEMPTED TO CONVINCE JOHN'S FRIEND THAT THAT FRIEND SHOULD LOAN
```

JOHN SOME REPAIR IMPLEMENTS.

[4]. HIS FRIEND SAID THAT HE WAS NOT ALLOWED TO LEND TOOLS.
JOHN'S FRIEND SAID THAT THAT FRIEND WAS NOT ALLOWED TO LOAN ANY REPAIR IMPLEMENTS.

[5]. JOHN ASKED HIS FRIEND TO SUGGEST SOMEONE FROM WHOM HE COULD BORROW TOOLS.
JOHN REQUESTED JOHN'S FRIEND TO RECOMMEND SOMEONE WHO JOHN COULD BORROW SOME REPAIR IMPLEMENTS FROM.

[6]. HIS FRIEND DID NOT ANSWER.
JOHN'S FRIEND DID NOT ANSWER.

[7]. FULFILLING HIS PROMISES WAS IMPORTANT TO JOHN.
DISCHARGING JOHN'S PROMISES WAS URGENT TO JOHN.

[8]. HE WAS ANGRY.
JOHN WAS ANGRY.

[9]. HE LEFT.
JOHN DEPARTED.

1.7 Thesis organisation

Chapter 2 of this thesis is a survey of anaphora in English; the emphasis is on characterising the anaphoric phenomena that must be dealt with, rather than at this stage attempting to define ways of dealing with those phenomena computationally.

The layout of subsequent chapters approximately parallels the stages by which SPAR processes a sentence, as outlined above.

Chapters 3 and 4 concern semantic representation and the lower-level semantic processing required for SPAR. The first of these chapters describes the preference semantics approach to representation from the point of view of anaphor resolution, while the second discusses SPAR's meaning representation.

We then turn our attention to the problem of using knowledge about focussing and discourse structure for anaphor resolution. Chapter 5 describes earlier attempts to solve this problem, especially that of Sidner, while chapter 6 presents the approach adopted in SPAR. Because the focussing component in SPAR constrains and to a certain extent controls the actions of the other components, chapter 6 also explores the problems of coordinating the different knowledge sources. Finally, the "collective heuristics" used when CSI is insufficient are described.

Next, world knowledge and its exploitation by inference are discussed. Chapter 7 presents the relevant background, particularly Wilks' work, while chapter 8 describes SPAR's Wilksian CSI mechanism.

The last component of SPAR to be examined is the paraphrase generator. Chapter 9 sets out the goals of the paraphrase process, discusses other relevant research and then describes SPAR's generator itself.

The last chapter, chapter 10, summarises and evaluates the system and suggests directions for further work.

Finally, there are appendices containing a selection of stories and the paraphrases made of them by SPAR. For some of these stories, part of the commentary SPAR makes on its own processing is also included.

2. Anaphora in English

"The material thing which is before you, that is It."

Huang Po (16th century Zen master)

A proper understanding of the problem of resolving anaphors by computer depends on an appreciation of the range of linguistic phenomena encompassed by the term "anaphora". This chapter therefore surveys the different types of anaphor that occur in English and examines some of the accounts of anaphora that have been given by linguists. I will not at this stage consider the *process* by which anaphors may be resolved; rather, I will analyse the *properties* - syntactic, semantic and pragmatic - which characterise anaphora and with which any attempt to resolve anaphors must come to terms.

First, in section 2.1, fundamental terms such as "anaphor",[1] "antecedent" and "text" are defined. Following Sidner [1979], the idea of specification,[2] rather than the more common idea of reference, is proposed as the basis for analysing anaphora. Anaphors are seen as *specifying*[3] elements in the (human or computational) hearer's or reader's conceptual representation of the text, rather than (necessarily) *referring to* objects in the world.

Section 2.2 is a descriptive survey, based on the taxonomy of Halliday and Hasan [1976], of various anaphoric phenomena that occur in English. In section 2.3, some theoretical analyses of the semantics of anaphora are discussed. Such analyses are often used in attempts to define configurational (typically syntactic) constraints on coreference between two noun phrases. Some of these attempts are therefore reviewed in section 2.4.

---

[1]From this point on, typographical conventions will be defined in footnotes when first used. Double quotation marks are used in three ways:

(1) to refer to words or phrases, often those in example texts;

(2) when discussing others' work, for terms which are either not used elsewhere in this thesis or are used in a different sense; and

(3) with the connotation of "so-called" or "supposed".

[2]Underlining is used when a new term is introduced or when it is properly defined for the first time. It is also used in example texts to highlight a particular word or phrase; in such cases, it carries no implication of any linguistic stress or emphasis.

[3]Italics are used for emphasis in the normal way.

2.1 Some basic terms and assumptions

Natural language is an efficient means of communication because of the assumptions that the speaker and hearer[4] can make about each other. The hearer assumes that what the speaker is trying to communicate forms a coherent whole rather than a set of isolated and unrelated comments; and the speaker assumes that the hearer is making such an assumption and is therefore trying to interpret what he hears as a connected message. This allows the speaker to use abbreviated linguistic forms, in the hope that the hearer will recognise them as such and interpret them with reference to what has already been said and the context of situation. The resulting piece of language exhibits the phenomenon of <u>cohesion</u> (in Halliday and Hasan's [1976] sense): i.e. there are semantic relationships between its parts which maintain its connectedness and which are independent of grammatical structure. A cohesive piece of language, produced by one or more speakers in a given context of situation, will be called a <u>discourse</u>. A written discourse produced by a single writer, who intends it to be interpreted without reference to any particular context of situation, will be called a <u>text</u>. This distinction between the terms "discourse" and "text" is not a standard one, but since the stories processed by SPAR are all "texts" in my more specialised sense, a separate term will be helpful.

The phenomenon of cohesion can be illustrated by the following contrasting examples.

(2-1) Handelsman is a great cartoonist. In a world currently blessed with good cartoonists, he stands skull and crossbones above the rest.[5]

(2-2) Handelsman is a great cartoonist. Before the different cases are considered, the methodology used for them must be stated explicitly.

Example (2-1) is a text; it discusses Handelsman and his stature as a cartoonist. Its textuality is exhibited by the cohesive relationships between the words "cartoonist" and "cartoonists", between "Handelsman" and "he", and between "Handelsman", "good cartoonists", and "the rest". Example (2-2) is not a text; its component sentences were taken from unrelated documents, and there are no cohesive ties between items in different sentences (although the habit of assuming textuality is so strong that the reader may detect cohesion where none was intended).

<u>Anaphora</u> is the special case of cohesion where the meaning (sense and/or reference) of one item in a cohesive relationship (the <u>anaphor</u>) is, in isolation, somehow vague or incomplete, and can only be properly interpreted by considering the meanings of the other item(s) in the relationship (the <u>antecedent(s)</u>). In example (2-1), "he" is an anaphor with antecedent "Handelsman", with which it shares what is often called "identity of

---

[4] I will use the terms "speaker" and "hearer" to include the writer and reader of written language.

[5] The beginning of Alan Coren's introduction to "Freaky Fables" by Handelsman, Sphere Books, 1979.

reference". "The rest" is an anaphor with antecedents "Handelsman" and "good cartoonists"; this is because the meaning of "the rest" in isolation is something like "the members of a given set, excluding a given member or members", which is incomplete in that it does not determine the identity of the set and the excluded member(s). On the other hand, the cohesive link between "cartoonist" and "cartoonists" is not anaphoric, since neither term depends on the other for its interpretation.

This initial characterisation of anaphora will be fleshed out as the different types are discussed in section 2.2. However, three points should be noted:

(1) My definitions of anaphor and antecedent are broader than many in AI, where some kind of identity (rather than just relatedness) of meaning is often required, and much broader than the use of the term by some linguists who use it to describe only forms such as reflexive pronouns which relate syntactically to their antecedents. My definitions are motivated by the fact that SPAR is geared towards resolving anaphors in the broad sense.

(2) Anaphora is sometimes distinguished from <u>exophora</u> and <u>cataphora</u>. Exophora is abbreviated reference to things in the nonlinguistic context of situation; we will not be concerned with it. Cataphora ("pointing forwards") is where the "antecedent" occurs later than the "anaphor". However, in this thesis I will use the term "anaphora" to subsume cataphora (but not exophora) except where otherwise stated.

(3) Anaphora (in my sense) is to do with vagueness rather than ambiguity; although in

(2-3) The shepherd saw a sheep stuck in a bramble bush.
He went to fetch a crook.

the correct interpretation of "crook" (as stick and not criminal) depends on its cohesive ties with "shepherd" and perhaps "sheep", these ties are not anaphoric because "crook" has, intuitively, two complete meanings rather than one incomplete one, and its interpretation involves selecting an already-present meaning rather than completing a partial one. (Of course, the line between vagueness and ambiguity cannot be drawn sharply, and so the presence or absence of anaphora will not always be so clear).

So far, cohesion and anaphora have been discussed purely as relationships within a text; but there is clearly more to them than that. It is conventional to say that the nature of the relationship between "he" and "Handelsman" in (2-1) is that they both <u>refer</u> to the same person, the cartoonist Handelsman. In the interests of readability, I will sometimes use the term "reference" informally, to describe the relationship between a phrase and what it "points to" (if anything) in the world described by the text. I will not use it to describe the relationship between anaphor and antecedent, as e.g. Halliday and Hasan [1976] do. This relationship will sometimes be called <u>coreference</u>.

However, the notion of reference is in some ways problematical. If the text in question is a fictional one (as all SPAR's texts are) it is not immediately clear to what sort of entities, if any, the phrases in it refer; and even if the status of those entities is clarified, it is hard to see what practical relevance they have to the working of a computational anaphor resolver. We will therefore characterise anaphora in terms of the notion of (co-)specification (Sidner [1979], appendix 2), which is adequate for present purposes. If we view the process of understanding a text as building up a model or representation of what is being discussed, then a piece of language such as a noun phrase can be viewed as specifying a single node or element of the model (and creating it if it does not already exist[6]). This element is the phrase's specification. Thus when (2-1) is understood, the word "Handelsman" is understood as specifying the model element which represents Handelsman (creating such an element if the reader has not previously heard of him); and "he" will be understood as specifying the same element. In other words, "he" and "Handelsman" cospecify. The status of these relationships of specification and cospecification is independent of Handelsman's status as a real individual; however, informally we will say that "he" and "Handelsman" refer to (the person) Handelsman.

This set of relationships is summarised in figure 2.1.[7]

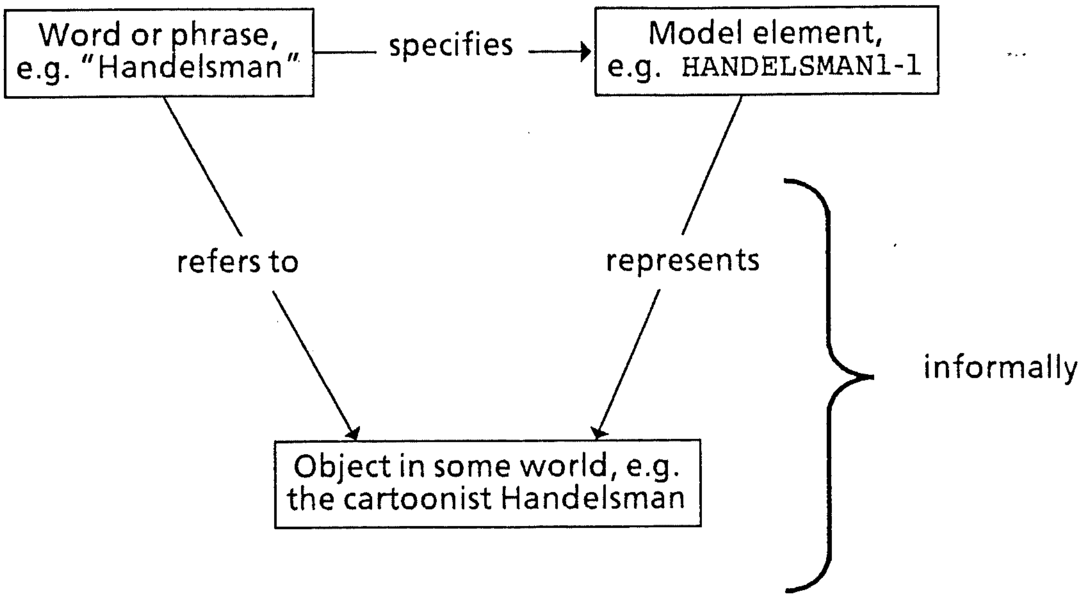


Figure 2.1: Specification, representation and reference

An object like a noun phrase might be viewed as introducing elements into the model in addition to the element it specifies. For example, the use of the noun phrase "a car" in a text might be seen as introducing not only its own specification, representing the car itself, but also other elements representing related concepts such as the car's driver, the action of driving the car, and

[6]Strictly speaking Sidner would not regard a term that resulted in the creation of a model element as specifying that element, but it seems to me that doing so leads to a more straightforward exposition without changing the content of her account.

[7]Following a convention used by SPAR, model elements will be denoted by objects such as HANDELSMAN1-1.

the generic or prototypic concept of cars.

2.2 Anaphoric forms in English

Having defined our basic framework, we will now make a descriptive survey of English anaphora, adopting the comprehensive framework of Halliday and Hasan [1976].

Halliday and Hasan distinguish five types of cohesion between items in a text: cospecification,[8] substitution, ellipsis, conjunction and lexical cohesion. These five types will be introduced by their occurrences in the following example, after which some of them will be discussed individually in more detail, and finally analysed within our specification-based framework.

(2-4) [1] One day John found his Jaguar wouldn't start.
[2] He asked Bill to come and look at it.
[3] Bill didn't really want to, but he came anyway.
[4] He saw that the plugs were worn, and advised John to fit some new ones.
[5] John did so.
[6] After that, the car worked perfectly.

Cospecification, as we have seen, is the relationship of identity of the things being talked about; an example is the cohesive link between "he" in [2] and "John" in [1], which identify the same individual.

Substitution, in contrast, operates at the level of sense; for example in the link between "some new ones" and "the plugs" in [4], "ones" picks up the sense of "plugs" but the two sets of plugs are distinct. Similarly, "did so" in [5] substitutes for "fit some new [plugs]" in [4].

Ellipsis is the special case of substitution by zero; the empty string after "didn't really want to" in [3] has an elliptical link to "come and look at it" in [2].

Conjunction is "a specification of the way in which what is to follow is systematically connected to what has gone before" (p227). In [6], "after that" signals the conjunctive relation of succession in time and perhaps also that of result.

Lexical cohesion is the use of repeated or semantically related words; examples are the links between "come" and "came", "Jaguar" and "car", and also, on a broad definition of semantic relation, between "Jaguar" and "plugs".

Of these five types of cohesion, cospecification, substitution and ellipsis are clearly anaphoric. Lexical cohesion can have an anaphoric aspect to it, according to our definition of anaphora; in (2-4), the semantic relation between "plugs" and "Jaguar" must be used to identify "the plugs" as being those of the Jaguar. Conjunction is not anaphoric; it corresponds with what

[8]Halliday and Hasan's own term is "reference", a term which we have decided to use only informally and then in a quite different sense.

Hobbs [1979] and others have called "coherence relations", which will be discussed in chapter 5. In the remainder of this section, English forms exhibiting each of the four anaphoric types of cohesion will be discussed in turn. The discussion uses the general framework of Halliday and Hasan; however, the examples originate elsewhere.

2.2.1 Cospecification

Halliday and Hasan distinguish three types of cospecification: personal, by means of function in the speech situation; demonstrative, by means of location, on a scale of proximity; and comparative, which is specification by means of identity or similarity. Comparative specification is beyond the scope of this thesis and will not be discussed.

(a) Personal cospecification is the type of anaphora to which most attention has been directed in linguistics and artificial intelligence. It is exhibited by definite personal pronouns such as "I", "it" and "they". It is worth noting in passing that such pronouns are more properly called "pro-NPs", since syntactically they usually behave as noun phrases and not as nouns; also, "it" can be used as a dummy subject filler. However, I will use the word "pronoun" in the conventional way.

Normally, a personal pronoun must have an explicit antecedent in the text, with which it agrees in gender, person and number. But neither explicitness nor agreement is essential, e.g. respectively

(2-5) Blend a cup of flour with some butter. Moisten it[9] with some milk, then knead it into a ball. (Webber [1978a])

(2-6) The most important qualification for the new programmer I want to hire is that they be fluent in Cobol. (Hirst [1981]).

A plural pronoun such as "they" (or, indeed, a plural non-pronominal noun phrase) may cospecify several items in the text which have not been mentioned in syntactic construction:

(2-7) When Ross visited his aunt Cicely, they spent the afternoon talking. Then, as arranged, Nadia arrived. Ross kissed his aunt goodbye, and set off with Nadia to the discotheque, where they danced the night away. (from Hirst [1981]).

In such cases, the problem of deciding which elements are included in the specification is non-trivial, involving syntactic, semantic and pragmatic factors.

The pronoun "it" can have a wide range of referents; as well as non-human objects, it can also, according to Webber [1978a], refer to specific events, event types, and propositions:

(2-8a) John dunked Mary's braids in the inkwell. It made her cry. (Specific event)

---

[9] As already stated, underlining in an example text is intended only to highlight a phrase under discussion, and does not indicate any linguistic stress.

(2-8b) John dunked Mary"s braids in the inkwell. Although it usually made her cry, today she held back. (Event type)

(2-8c) To prove that all cats have three legs, let's assume its converse. (Proposition)

But not all anaphoric uses of "it" and "they" are cospecifying ones; some can be classed as substitution, and are discussed below.

Reciprocal pronominal phrases such as "each other" and reflexive pronouns, such as "herself", take antecedents only from the sentence in which they occur, and are not covered by Halliday and Hasan, who are only concerned with cohesion between sentences. They differ from personal pronouns mainly in their syntactic behaviour, as discussed below in 2.4.

(b) Demonstrative cospecification is, for Halliday and Hasan, exhibited by words such as "this", "these" and "here" (for "near" items), "that", "those", "there" and "then" (for "far" items), and "the" (with neutral proximity). Many of these words can act either as modifiers ("these words") or as heads ("Think about this."). The exact meaning of "near" and "far" in the classification of demonstratives is complex, involving spatial and temporal aspects (both in the world and in the text) and the degree of association with the speaker.

Definite noun phrases introduced by "the", although classified by Halliday and Hasan as demonstratives, will be discussed below under the heading of lexical cohesion.

2.2.2 Substitution and ellipsis

Substitution and ellipsis are anaphoric devices by which anaphors derive from their antecedents not their specification but their sense. Such anaphors will be called descriptional (after Webber). Since ellipsis is simply "substitution by zero", differing from substitution only grammatically and not cohesively, the two phenomena may be discussed together.

Halliday and Hasan distinguish between nominal, verbal and clausal varieties of substitution and ellipsis. "Nominal substitution" means not substitution *for* a whole noun group or noun phrase but substitution of one or more items *in* it, and similarly for the other five categories. Verbal and clausal ellipsis and clausal substitution will not be discussed here, because they usually result in sentences which Boguraev's analyser, and hence SPAR, cannot deal with.

(a) Nominal substitution is, for Halliday and Hasan, exhibited mainly by some uses of the English words "one" and "ones". When used as substitutes (rather than alone in a noun phrase, which is ellipsis) "one" and "ones" serve to contrast items. The contrast may be explicit:

(2-9) Wendy prefers the yellow T-shirt to the red one.

or it may serve to pick out one member of a set:

(2-10) Of her two Dior T-shirts, Wendy prefers the yellow one. (from Webber [1978a]).

where by implication one T-shirt is yellow and the other is not.

(b) Nominal ellipsis occurs when the head of a noun phrase (along with perhaps other elements) is deleted, and is replaced by another element, which may take a slightly different form. For example, whereas (2-9) exhibits substitution, (2-11) exhibits ellipsis:

(2-11) Wendy prefers the yellow T-shirt to the red.

In (2-9), the head noun "T-shirt" is substituted by "one"; in (2-11) it is deleted altogether, and the modifier "red" acts as the head of the noun phrase.

When determiners and possessive pronouns move into head position, they may undergo a change of form. "A" and "an" become "one", and "your" becomes "yours", other possessive pronouns changing similarly. Examples of such ellipsis are:

(2-12) Wendy found the T-shirts so irresistible that she stole one.

(2-13) I prefer my T-shirt to yours.

The underlined pronouns can be regarded as derived by ellipsis of the head noun from the noun phrases "a T-shirt" and "your T-shirt(s)" respectively. Although the elliptical and substitutionary uses of "one" may appear to be similar, they are distinguished by the fact that substitute "one" behaves syntactically as a noun, whereas the elliptical "one" behaves as a noun phrase.

Halliday and Hasan's treatment may be extended to phrases with the determiner "the" and to those without any determiners or possessives. Just as "a" becomes "one", "the" becomes "that" or "those" depending on whether the deleted noun is singular or plural:

(2-14) The population of China is higher than that of Russia;

and when there is no determiner, and ellipsis disposes of every element in the noun phrase, a pronoun such as "it" or "they" is brought in to prevent the noun phrase being deleted altogether (noun phrase deletion does happen, but is governed by the rules for verbal and clausal ellipsis). This view explains so-called "pronouns of laziness" in paycheck sentences:

(2-15) The man who gave his paycheck to his wife was wiser than the man who gave it to his mistress.

where common sense reasoning tells us that "it" describes the second man's paycheck rather than the first, and is therefore a descriptional anaphor; and also this example from Webber [1978a] (which Webber in fact classifies as generic reference):

(2-16) A Rhodesian ridgeback bit me yesterday. They are really vicious beasts.

(c) Verbal substitution occurs when items within a verb phrase are deleted. It is marked in English by the verb "do", possibly followed by "so", "it", "this" or "that". These forms all have approximately the same meaning, although "do" on its own differs from the other forms in its syntactic behaviour.

2.2.3 Lexical cohesion

Lexical cohesion is exhibited by the presence of identical or semantically related lexical items in a text. It is essential to make use of it in resolving anaphors containing significant intrinsic lexical information, such as full (i.e.

non-pronominal) definite noun phrases (henceforth FDNPs).

Halliday and Hasan distinguish between reiteration, where lexically cohesive items occur in cospecifying phrases, and collocation, where the relationship is less direct.

(a) In the case of reiteration, an approximate information constraint usually operates: that is, the anaphoric lexical item must usually contain no information that its antecedent lacks. This is because the function of an anaphor is typically to access given information rather than to introduce new. So in the following,

(2-17) John has bought a new car. The (car/vehicle/thing/?Jaguar)[10] goes at quite a speed.

the use of "the Jaguar" is odd in a neutral context, since it introduces new information while being marked, through the definite article, as anaphoric. The reader may fail to identify it with the "new car", and expect a continuation like "...but the new car's really slow".

In fact the information constraint applies to noun phrases in context rather than single lexical items; so

(2-18) A man came up behind John and hit him on the head. John turned round to face his assailant.

is acceptable since the information that the man is an assailant is inferable (i.e. not new) when "his assailant" is read. (Contrast the case where "and hit him on the head" is missing).

An apparent exception to the information constraint is the use of epithets:

(2-19) John has bought a new car. The idiot is always overspending.

Here, however, the extra information concerns the speaker's opinion of the facts rather than the facts themselves.

Another exception occurs when new information serves to select one or more elements of an already-mentioned set:

(2-20) John has bought several new cars. The Jaguar is particularly impressive.

Furthermore, a text may still be odd even when the information constraint is satisfied:[11]

(2-21) ℵ[12] John has bought a new car. The mammal is very pleased with it.

The oddness here is perhaps because the fact that John is a mammal may not

---

[10] Items in parentheses separated by slashes indicate alternatives. A question mark indicates doubtful acceptability under the interpretation in question.

[11] I am grateful to Peter Bosch for pointing this out.

[12] A "ℵ" is used to indicate a text which is in some way odd even though every sentence in it is acceptable on its own.

be very easily retrievable; "mammal" is a specialised word for many people.

(b) Collocation, while not always anaphoric, can sometimes be used to identify objects whose existence is suggested by the context but which have not actually been mentioned. In

(2-22) John has bought a new car. The indicators use the latest laser technology.

the collocation between "car" and "indicators" is the linguistic counterpart of the real-world relationship between cars and indicators.

The information constraint applies to anaphoric collocation as well as to reiteration, although less strictly. In the above example, although the mention of the indicators is new, the fact that the car has them is not. Texts such as the following (from Sidner [1979])

(2-23) The heiress lived the life of a recluse. She died under mysterious circumstances, but the murderer was never found.

are acceptable, since the phrase "died under mysterious circumstances" makes murder highly probable although not certain. The text becomes progressively odder if "under mysterious circumstances" is altered to "ten years ago" and then to "in hospital", since these phrases make murder progressively less likely.

Even when collocation is not anaphoric, it may still help in interpretation. Consider

(2-24a) John's going to lose this trick. He hasn't got a jack.
(2-24b) John will never get that wheel off. He hasn't got a jack.

where the collocation between particular ***senses*** of the words "trick" and "jack" on the one hand, and "wheel" and "jack" on the other, leads to the correct interpretation of the ambiguous word "jack".

2.2.4 Cohesion and semantic representation

Some light can be shed on the relationships between the types of cohesion discussed above by comparing their roles in SPAR's semantic representation scheme, which we looked at briefly at the end of chapter 1. SPAR maintains a pair of networks, each consisting of nodes linked by various semantic relationships. The static word sense network (WSN) has nodes representing word senses, and the incrementally constructed text model network (TMN) has nodes which are specified by phrases in the text. Instance or "is-a" links join TMN nodes to WSN nodes which are their "descriptions", and the same set of semantic relationships is used within both networks. We will see below that relationships in one network can be paralleled by relationships between corresponding nodes in the other.

In such a model, cospecification corresponds to the simple case of anaphor and antecedent specifying the same TMN element. Substitution and ellipsis correspond to a relationship between specifications which is mediated by links to shared WSN nodes; anaphor and antecedent may specify different TMN elements, but the descriptions of these elements have something in common. Thus in (2-4), the element specified by "the plugs" is distinct from that specified by "some new ones", but the two elements both have links to the WSN node for (the relevant sense of) "plug".

Lexical cohesion corresponds to a relationship between WSN nodes which may or may not be parallelled in the text model. In (2-4), the lexical cohesion between "plugs" and "Jaguar" might be expressed in the WSN by links stating that a Jaguar is a type of car and a (spark) plug is part of a car. These links correspond (for (2-4)) to analogous ones in the TMN indicating that these specific worn plugs are part of this specific Jaguar.

To summarize: cospecification is a relationship entirely at the TMN level; substitution and ellipsis operate indirectly, via the WSN; while lexical cohesion operates at the WSN level and sometimes also, in parallel, in the TMN. Thus different anaphoric relations are modelled in SPAR by different methods of representation.

2.3 How anaphors relate to their antecedents

Although most work on anaphora in AI has been concerned with selecting appropriate referents or specifications from a number of possibilities, a logically prior problem is that of identifying what the text makes available as possible specifications. In some AI systems it is tacitly assumed that an antecedent noun phrase makes available only the specific individual it refers to in the world described by the text. This approach is not in general adequate, as can be seen from the following texts. In none of these do the pronouns refer to single specific individuals.

(2-25) Every farmer owns a donkey. <u>He</u> beats <u>it</u>.
(2-26) No child will admit that <u>he</u>'s sleepy.
(2-27) The lion is a large mammal. <u>It</u> is found all over Africa.

This section reviews some of the accounts of such phenomena that have been offered. Following an overview of accounts within theoretical (non-computational) linguistics, I shall review the more computationally oriented approaches of Kamp [1981] and Webber [1978a,b].

2.3.1 An overview of work on the semantics of definite anaphora

Many attempts have been made to account for the semantics of definite anaphora, and it is impossible to do justice here to the very complex arguments involved. Instead, I will briefly summarize some of the phenomena which have been seen as problematic and outline the kinds of accounts of them that have been offered.

In early transformational theory (e.g. Lees and Klima [1963]), a "pronominalisation transformation" was postulated in which one occurrence of a repeated noun phrase was optionally replaced by a pronoun; thus both of

(2-28a) John threw the ball and Mary caught the ball.
(2-28b) John threw the ball and Mary caught it.

were seen as being derived from the same deep structure. On this view, all anaphoric pronouns were regarded as pronouns of laziness (see (2-15)). This treatment is inadequate in many ways, failing to account, for example, for Bach-Peters sentences such as

(2-29) I gave the book that he wanted to the man who asked for it.

where each pronoun is contained in the other's antecedent phrase.

Since the idea of a pronominalisation transformation was abandoned, authors attempting to define a semantics for definite anaphora have typically first distinguished a number of categories of such anaphora, and then attempted to show that these categories either can or cannot be accounted for in the same way. See, for example, Partee [1972,1978]; Stenning [1978]; Bosch [1980]; Evans [1980]; Kempson [1983].

Broadly speaking, the most widely-accepted category distinction, and the one to which most attention has been devoted, is between the following two types of pronoun:

(1) Bound variable pronouns (Partee [1978], Reinhart [1983]; cf also Bosch's [1980] roughly coextensive category of "syntactic" pronouns). These are often thought of as restricted to occurrences in syntactic construction with their antecedents, and are fully interpreted at the level of semantics, typically by representing pronoun and antecedent by the same bound variable in the logical form. An example is the following sentence with its accompanying logical form.

(2-30) No boy wants Mary to dislike him.
(L2-30) $\forall x$ (Boy(x) $\Rightarrow$ Not(Wants(x,Dislike(Mary,x))))

(2) Pragmatic pronouns (Partee [1978]; also called "discourse" pronouns by Kempson [1983], and "referential" pronouns by Reinhart [1983]; cf Bosch's [1980] similar category of "referential" pronouns) which require pragmatics as well as semantics for their interpretation. For example:

(2-31) The postman gave John a letter. He tore it open.

It is arguable that the same account can be given for pragmatic pronouns whether they are anaphoric or exophoric; in other words, that when a pragmatic pronoun has an antecedent, they relate only indirectly and secondarily, by way of their shared referent (or specification). This is in contrast to bound variable pronouns, which are anaphors directly related to their antecedents by syntactic agreement.

It is often pointed out that a pronoun may be ambiguous between the bound variable and pragmatic categories on a particular occasion of use. Such ambiguity does not affect the truth conditions of the sentence in which it occurs, but may affect those of later sentences. Consider for example

(2-32) John beats his wife. The Governor of New Hampshire does too.

If "his" is treated as a bound variable anaphor, the anaphoric "does" will be interpreted as "beats *S*'s wife" where *S* is the subject of "does", the Governor. If "his" is treated as pragmatic, "does" will be interpreted as "beats John's wife".

Some types of definite anaphora do not seem to fall into either the bound variable or the pragmatic category. These are often thought to include the pronouns of laziness mentioned in section 2.2.2 above. However, in 2.2.2, it was argued that pronouns of laziness are in fact substitutes; that is, they are cohesive at the level of sense rather than specification. If so, it is unsurprising that they are a problem for accounts of definite anaphora.

Another difficult category is that of pronouns in <u>donkey sentences</u> such as

(2-33) Every farmer who owns a donkey beats <u>it</u>.

Although "it" here appears at first sight to be a bound-variable pronoun, closer examination reveals that under the most natural interpretation, it falls outside the scope of the "donkey" variable and therefore cannot be bound by it:

(L2-33) $\forall$ f {(farmer(f) & $\exists$ d [donkey(d) & owns(f,d)]) $\Rightarrow$ beats(f,<u>?it</u>)}

Having set out the kind of problems that arise, we will now summarise the conclusions of some of those who have attempted to solve them. The arguments on which those conclusions are based are often quite involved, and so for the sake of brevity they are not reproduced here.

Partee [1972] considers bound variable pronouns, pragmatic pronouns and pronouns of laziness, and attempts to outline a uniform approach. She concludes that while pragmatic pronouns could perhaps be accounted for by an extended bound variable treatment, pronouns of laziness are more difficult. However, in a later paper [1978], she takes the view that the bound variable approach is less comprehensive than had earlier seemed likely, and cannot be extended to pragmatic anaphora. On the other hand, she argues that both pronouns of laziness and "donkey" pronouns (but not bound variable pronouns) can be accounted for as an extension of pragmatic pronouns, using the framework of Cooper [1979].

Stenning [1978] also develops an extended pragmatic approach, which he applies to a range of bound variable anaphora. However, in this regard Kamp's work (see 2.3.2 below) is perhaps more impressive.

Kempson [1983] questions the traditional bound-variable/pragmatic distinction, arguing, with examples, that "every phenomenon which indicates the pragmatic nature of discourse anaphora [what we have called "pragmatic anaphora"] is displayed also by bound-variable anaphora", and that therefore a pragmatic basis must be assumed for "bound variable" anaphors as well as "pragmatic" ones.

Wiese [1983] goes further than Kempson, taking the view that anaphora is a fundamentally pragmatic phenomenon, and that the purely linguistic notion of antecedence, on which bound-variable and some pragmatic approaches rely, is not helpful in accounting for it. He argues convincingly that a pragmatic account in terms of speaker reference is adequate for (so-called) bound-variable as well as "pragmatic" anaphora, and sketches how bound variable-anaphora could be dealt with by basing the semantic interpretation process on Hintikka's game-theoretical semantics (Hintikka and Carlson [1979]).

There are thus reasons to believe that a unified, probably pragmatically-based, explanation for definite anaphora of all types does exist. However, even if this is the case, it does not follow that one should design a language *processing* system to exploit it, because syntactic and semantic processing is much better understood than pragmatic. Thus a treatment which ascribes a significant syntactic and/or semantic role to anaphor resolution may be preferable for practical natural language processing even if its predictions are only approximately correct.

The various approaches discussed nevertheless provide helpful insights into the complex nature of anaphora and the difficulty of fully characterising it. However their usefulness to the present project is limited by the fact that they all attempt to describe only the circumstances under which a statement is true and not the process by which it can be understood. This is less true of the work of Kamp and of Webber, which will now be discussed in turn.

2.3.2 Discourse Representation Theory

Kamp's discourse representation (DR) theory is an approach to anaphora which attempts to address the concerns of both linguists and AI workers. Kamp [1981] presents a grammar for a small fragment of English and a set of DR formation rules which act on syntactic analyses of text sentences derivable from the grammar. The formation rules construct a discourse representation structure (DRS), which is an implicitly structured set of DR's which can be given a model-theoretic interpretation. A single DR consists of a number of elements taken from the universe of the representation (for which mappings can be defined into a model) and a number of constraints on those elements and elements defined in other DR's dominating the first in the DRS. As an example, the DRS for the "donkey sentence" (2-33), repeated here,

(2-34) Every farmer who owns a donkey beats it.

would be as in figure 2.2 (drawn according to Guenthner and Lehmann's [1983] conventions rather than Kamp's own).

In model-theoretic terms, a DRS is true if there is an embedding (assignment to model elements) of variables bound at the outermost level which satisfies the accompanying constraints, and for all its substructures of the form "A ⇒ B", every embedding for which A is true can be extended (by assigning values to any variables bound in B) to one in which B is also true. Thus the DRS in figure 2.2 may be read "For every assignment of values to x and u such that x is a farmer, u is a donkey and x owns u, it is true that x beats u". In contrast to some of the "bound variable" treatments reviewed above, "donkey" pronouns can therefore by treated in the same way as "bound variable" anaphora (although the bound-variable/pragmatic distinction is not one that Kamp makes).

Kamp's fragment of English, although small, is non-trivial in that it includes the determiners "a" and "every", conditionals (if...then), relative clauses, and the personal pronouns "he", "she" and "it". Other work (e.g. van Eijck [1984], Klein [1984]) has explored possible extensions of DRT to other determiners (especially plurals) and verb phrase ellipsis. Kamp's DR formation rules are not complete where pronoun interpretation is concerned, since they assume a

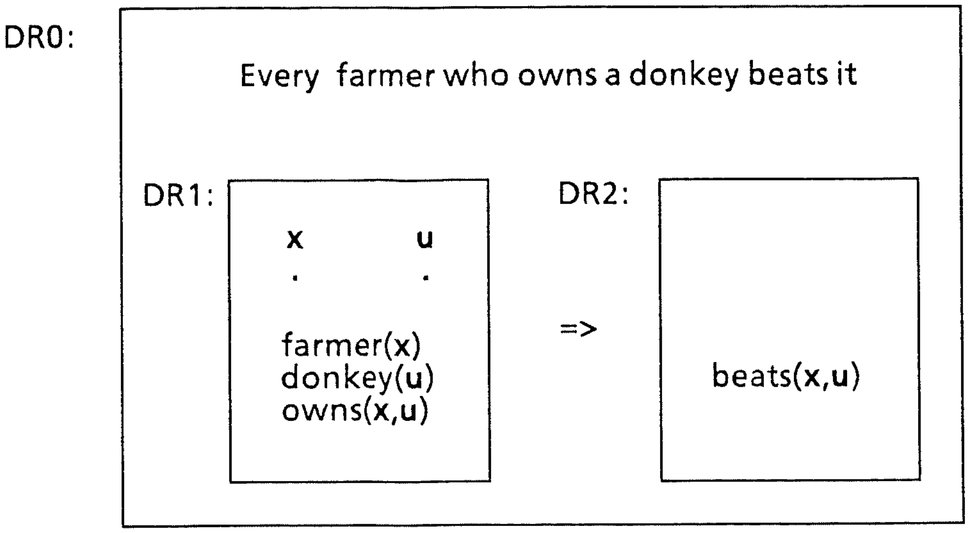


Figure 2.2: Discourse representation structure for "Every farmer who owns a donkey beats it"

mechanism for choosing a "suitable" referent from among those accessible[13] in the DRS; the importance of Kamp's work from the point of view of anaphor resolution is that it provides the basis of a particularly powerful uniform treatment of anaphora (by means of the binding of variables in DR's below top level) and a useful constraint on coreference, which explain the unacceptability of sentences such as

(2-35) If Pedro owns every donkey then he beats it.

in which the element representing the (prototypic) donkey is inaccessible from the DR in which "it" must be resolved. This constraint is quite different from configurational constraints such as Reinhart's (discussed below) because it operates at the level of logical relations.

Kamp's is probably the best unified treatment of anaphora developed to date, because of its relative simplicity and the way in which it is correlated explicitly with a small but significant fragment of English. Given the mechanism of constructing subsidiary DRs, "pragmatic" and "bound variable" anaphors are treated in exactly the same way in that the DR formation rules make no distinction between them. In addition, Guenthner and Lehmann [1983] have shown in the context of relational database query that Kamp's theory, particularly the accessibility constraint, can make a useful contribution to computational anaphor resolution.

[13] A referent is accessible if it is bound in the DR where the pronoun must be assigned a referent or in a superordinate DR. DRα is superordinate to DRβ if DRα contains DRβ or if DR$\alpha \Rightarrow$ DRγ, where DRγ is or contains DRβ.

2.3.3 Webber's formalism

Webber [1978a] made significant steps towards achieving two aims: that of identifying what the text made available to anaphora, and that of developing, for NLP purposes, a formalism for representing sentences, and procedures operating on that formalism, so as to guarantee that the correct specification (or in Webber's terms, referent) for an anaphor was among those made available. She assumed the existence of procedures to select the correct specifications from among those available.

Webber examined definite pronouns, descriptional anaphors and elided verb phrases. A review of her treatment of each of these would be unreasonably long because of the complexity of her formalism; we will therefore only consider the her analysis of definite pronouns, which is arguably the most central to her approach.

For her formalism, Webber developed a sorted logic which allowed the representation of "discourse entities" (DE's; roughly, concepts which SPAR would represent as text model network elements) for individuals, sets, stuff, generics, prototypes, events, descriptions and predicates, all of which were viewed as possible specifications of anaphors. Each DE had at least one <u>invoking description</u> (ID), a logical expression which could be used in resolving a subsequent anaphor. IDs are exemplified below.

In the processing model she assumed, English sentences were parsed into a surface syntactic tree, which was transformed, using information in the lexicon, into a semantic representation in which considerable ambiguity remained. The derivation from this of a logical form (to be exact, an extended predicate calculus representation with restricted quantification) involved (1) deciding on quantifier scope, (2) choosing specifications of definite pronouns (or at least deciding whether a bound variable interpretation was possible), and (3) deciding whether FDNPs (full definite noun phrases) were anaphoric and if so replacing them by a referent label representing a discourse entity. If and when later anaphors required it, a set of <u>ID rules</u> could be applied to sentence logical forms to derive ID's to be used in anaphor resolution.

ID rule application may be exemplified as follows. The sentence

(2-36) I saw a cat.

gives rise to the logical form

(L2-36) $(\exists x : \mathrm{Cat})\ .\ \mathrm{Saw}\ I,x$

(read as "sentence L2-36 : there exists an entity x, which is a cat, such that I saw x"). There is an ID-rule which says that from a logical form matching the pattern

(Lj) $!(\exists x : C)\ .\ F_x$

where C is an arbitrary predicate on individuals and F_x is an arbitrary open sentence in which x is unbound, we can extract the ID

ιx: Cx & F_x & **evoke**(Lj,x)

(read as "the x such that Cx and F_x and such that sentence Lj evokes[14] x").
Thus from (L2-36) we can derive

ιx: Cat x & Saw I,x & **evoke**(L2-36 ,x)

(read as "the x such that x is a cat, I saw x, and sentence L2-36 evokes x"; or less formally, "the cat mentioned in sentence 2-36").

A more complex and interesting example is

(2-37) Each cat that Wendy owns dislikes Sam.

which has logical form

(L2-37) ($\forall$x : λ(u:Cat) [Own Wendy,u]) . Dislike x,Sam

(read as "For every x satisfying the complex predicate on u 'u is a cat and Wendy owns u', x dislikes Sam"). An ID rule in fact derives two ID's for the DE representing the cats: one for the "prototypical cat" and one for the set of cats:

x : λ(u:Cat) [Own Wendy,u]
ιx: **maxset** λ(u:Cat [Own Wendy,u]) x

to be read as "any x such that x is a cat that Wendy owns" and "the x which is the maximal set of entities satisfying the condition of being a cat that Wendy owns". The justification for deriving these two ID's is that each of them allows for one of the following alternative continuations:

(2-38a) <u>It</u> skulks in a corner when he's around.
(2-38b) He's not too fond of <u>them</u> either.

Webber's logical form and ID rules are motivated by, and represent adequately, a wide range of linguistic phenomena. These include definiteness, types of determiner, the bound-variable/pragmatic distinction, the possibility of distributive and collective interpretations for sentences involving plural noun phrases, and the way that elided verb phrases can create DE's which later pronouns can specify. (See Webber [1978a] for details).

The importance of Webber's work is her analysis of a wide range of anaphoric phenomena not previously considered in the processing context characteristic of AI, and her resulting "logical form" representation which is motivated primarily by the requirements of eventual anaphor resolution rather than by independent, non-linguistic criteria such as proof properties or completeness. Her formalisation of the problem and of her partial solution to it is impressive since it allows the claims and limitations of her proposals to be rigorously examined, something she is careful to do.

A particularly strong point of her scheme is that it allows a great deal of "problem-driven" ambiguity resolution; moreover, before the "problem" occurs, the ambiguity is (mostly) represented as a single, vague form rather than a number of alternative fully-determined ones. The only serious exception to this is in the area of quantifier scoping, where each logically distinct reading must be represented separately. See Hobbs [1983] for a discussion of the drawbacks of such a situation and a suggested solution.

---

[14] DE's are <u>e</u>voked by sentences and <u>in</u>voked by ID's. The difference in prefixes emphasises the difference in kind between these relationships.

The limitations of her work, many of which she points out herself, seem to me to arise mainly from the fact that she is looking only at a part of the anaphor resolution process: the process of determining what the text makes available to pronominal and verb phrase anaphora in later sentences. Intra-sentential anaphors are not easily dealt with if they must be resolved before they ID's they require are derived; Webber is aware of this problem and suggests the use of "vague, temporary IDs", but their introduction would probably sacrifice much of the elegance of her method. Also, it remains to be shown that her approach can be integrated into one that takes account of global discourse structure (both to constrain the set of currently available antecedents and to allow anaphoric references to larger structures such as sentences, paragraphs and chapters) and allows non-deductive reasoning such as that described in chapter 7. Similarly, it is not clear how easily her logic-based approach could support the resolution of non-pronominal anaphors, which as we have seen can relate to their antecedents in a wide variety of ways. Such phenomena seem to be more naturally catered for by, for example, a frame-based approach. In this context it is worth noting Charniak's [1981b] knowledge representation in which a frame-like organisation is imposed on a restricted range of predicate calculus statements, in order to combine the organisational advantages of the former with the logical rigour of the latter.

2.4 Configurational constraints on coreference

Considerable efforts have been made by linguists, usually working within a framework related to Partee's [1978], to state configurational (typically surface syntactic) constraints on coreference (more strictly, cospecification; however, the distinction is not crucial here). A number of different constraints appear to operate; they apply differently to R-pronouns (reciprocals such as "each other" and reflexives such as "himself") and non-R pronouns, and to definite and indefinite (quantified) antecedents. The following constraint, slightly reworded from rule (34) in Reinhart [1983,p158], appears to be approximately correct:

(2-39) In the surface syntactic structure of a sentence, an R-pronoun must have an antecedent dominated by its minimal governing category (MGC); a non-R pronoun may not have an antecedent dominated by its MGC.

Definition: The MGC of a node α is the minimal (i.e. lowest) S or NP node dominating α.[15]

A further constraint for non-R pronouns has received much attention. The position taken in this thesis is that of Reinhart [1983], who shows that her original (Reinhart [1976]) rule is inadequate. However, the superior (1983) version is best understood by presenting the earlier one first; also, it is the earlier rule which is used by SPAR, for reasons which will be presented.

---

[15]Strictly speaking, the MGC of α is the minimal S or NP dominating both α and its governor, where the governor of α is the node that "assigns case" to α, e.g. N, V or P (Reinhart [1983,p139]).

Reinhart's original rule is as follows.

(2-40) If NP1 c-commands (and is distinct from) NP2, and NP2 is not a pronoun, then NP1 and NP2 are non-coreferential.

Definition: A <u>c-commands</u> B if the branching node α_1 most immediately dominating A either dominates B or is immediately dominated by a node α_2 which dominates B, and α_2 is of the same category as α_1.

A "branching node" is a node with more than one daughter; by "category" Reinhart means the label of a node ignoring any bars it may have, so that for example S and $\bar{S}$ are of the same category.

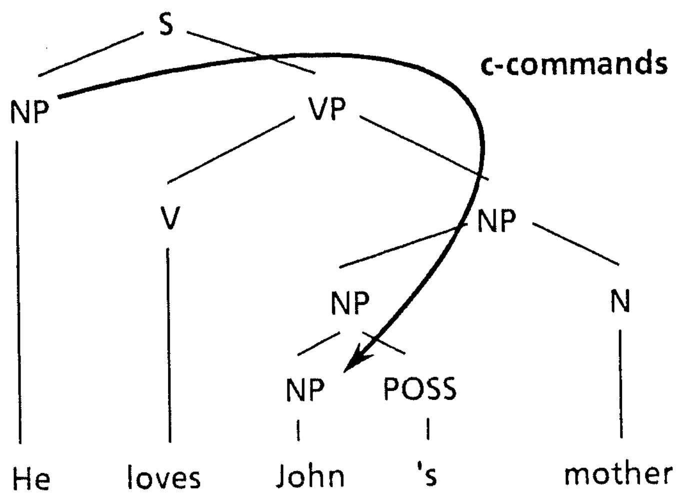
As a simple example of the application of (2-40), consider

(2-41) He loves John's mother.

which has the surface structure shown in figure 2.3. In this structure, the NP node for "he" c-commands that for "John"; Reinhart's rule therefore correctly predicts that "he" cannot corefer with "John". Note that the rule does not require NP1 to be a pronoun; thus coreference is also ruled out in

(2-42) John loves John's mother.

which is, if not unacceptable, at least odd in isolation.



<u>Figure 2.3: Surface syntactic structure showing c-command relation</u>

An alternative (and, we will see, inferior) structural relation to Reinhart's is that of Lasnik [1976]. Lasnik's rule can be stated as follows:

(2-43) If NP1 precedes and kommands NP2, and NP2 is not a pronoun, then NP1 and NP2 are disjoint in reference.

Definition: A <u>kommands</u> B if the MGC of A also dominates B.

Lasnik's rule predicts correctly for sentence (2-41). However, c-command performs better for

(2-44) Near Dan, he saw a snake.

for which it correctly rules out coreference while Lasnik's rule does not. Reinhart [1983] discusses a range of other cases in which only c-command performs correctly. We may therefore conclude that c-command is the more appropriate relation.

As well as (2-39) and (2-40), Reinhart argues for a stricter restriction for quantified (including indefinite) NPs:

(2-45) A quantified NP can be the antecedent of a pronoun only if it c-commands that pronoun.
(paraphrased from Reinhart [1983,p122]).

For example, whereas (2-46a) allows a coreference interpretation, (2-46b) does not:

(2-46a) If he turns up, tell John to wait outside.
(2-46b) If he turns up, tell an applicant to wait outside.

In (2-46a) neither "he" nor "John" c-command each other, so coreference is allowed by rule (2-40) while rule (2-45) does not apply. However, in (2-46b) "an applicant" is quantified (in the broader sense) and so (2-45) forbids coreference.

But Reinhart [1983] points out that her rule (2-40) has several types of apparent counterexample which suggest that no purely configurational approach to coreference restrictions can be entirely accurate.

The first such problem, which is presented by certain sentences containing preposed prepositional phrases (PPs), Reinhart disposes of by allowing pragmatic factors to influence the determination of syntactic structure. The structure a reader assigns to such a sentence depends on pragmatic factors; but once the structure is determined, coreference options are, Reinhart argues, determined entirely syntactically. Thus pairs such as (2-46a-b) are no problem for rule (2-40).

The second type of objection is more serious, and, Reinhart shows, necessitates a complete reworking of the framework in which the c-command rule operates. From a position similar to Partee's ([1978]; see section 2.3.1) she argues that although a syntactic rule may prevent two noun phrases being coindexed (i.e. being represented by the same variable at the level of logical form), it cannot prevent them taking the same referent anyway, since referent selection is a pragmatic rather than a syntactic process. Evans [1980] presents a number of contexts in which two NPs corefer in spite of breaking rule (2-40). For example,

(2-47) Everyone has finally realised that Oscar is incompetent.
Even he has finally realised that Oscar is incompetent.

is acceptable (though perhaps odd) if we interpret "he" as "Oscar"; it is far less odd if "he" is replaced by "Oscar" (still violating (2-40)). Evans suggests that rules such as (2-40) forbid not coreference but "referential dependency", but Reinhart criticises this notion as undefinable. Instead, she proposes that a pronoun may only be coindexed with an NP which c-commands it (with rule (2-39) also applying), and that coindexing results in a bound-variable interpretation (see section 2.3.1). This proposal explains why rules (2-40) and (2-45) both make use of the c-command relation, since quantified NPs can (in

Reinhart's treatment) only act as antecedents by means of the bound-variable mechanism and not by coreference.

To deal with the differences between

(2-48a) Zelda thinks that she is boring.
(2-48b) She thinks that Zelda is boring.

where coreference is possible in the former but not in the latter, Reinhart uses the Grice-derived principle that "if a speaker has the means to express a certain idea clearly and directly, he would not arbitrarily choose a less clear way to express it" (Reinhart [1983,p166]) and the observation that the bound variable mechanism, while not totally unambiguous, is more constrained and therefore more "clear and direct" than the coreference mechanism. She suggests the following strategies (trivially reworded):

(2-49) *Speaker's strategy:* Where a syntactic structure you are using allows bound-variable interpretation, then use it if you intend your expressions to corefer, unless you have some reasons to avoid bound-variable anaphora.

Hearer's strategy: If the speaker avoids the bound-variable options provided by the structure he is using, then, unless he has reasons to avoid using bound-variable anaphora, he did not intend his expressions to corefer.

Thus in a sentence of the form "x thinks that y is boring", a bound variable interpretation is possible if y is a pronoun, as in (2-48a); if the speaker uses (2-48b), the hearer assumes that coreference is not intended. The apparent correctness of rule (2-40) therefore arises from the fact that if its condition is satisfied (i.e. if NP1 c-commands a non-pronoun NP2) then the bound variable option could have been used (by making NP2 a pronoun) if coreference had been intended.

The third type of problem for Reinhart's approach is represented by a mixed bag of sentences which violate (2-45) in a way which can only be repaired, if at all, by ad-hoc modifications to the theory. However, Reinhart claims that any configurational approach, whether syntax- or semantics-based, would suffer from the same problems. Other problems for configurational approaches are discussed by Bolinger [1979], but it is likely that many of them can be solved by allowing pragmatic factors to operate indirectly at appropriate points, as Reinhart does for the preposed PP problem and in (2-49), while maintaining the purely configurational nature of the constraints.

Thus although Reinhart's account is not perfect, it seems to be the best yet offered, and fails in only a few cases, none of which SPAR is likely to encounter. It therefore makes available to SPAR a useful, reliable and computationally cheap source of knowledge. The approach taken in applying it is, as in other parts of the system, a shallow processing one. It is assumed, as is normally the case, that the writer of the story has no "reasons to avoid using bound-variable anaphora"; rules (2-40) and (2-45) may therefore be applied. In other words it is effectively Reinhart's earlier (1976) rule that is used. Adjustments necessitated by the fact that SPAR has no direct access to surface structure will be discussed in section 6.4.1.

3. Preference Semantics

> "Plausible impossibilities should be preferred to unconvincing possibilities".
>
> *Aristotle, Poetics 24*

In order to resolve anaphors reliably, a language processing system must be able to do at least two things: to represent the specifications that an anaphor might have, and to select the intended specification from among those available. In chapter 2, we looked at some of the phenomena of which a solution to the first of these two problems must take account.

Wilks' theory of preference semantics, which significantly influenced the design of SPAR, can be viewed as an attempted solution both to the problem of representation and the problem of selection. Wilks developed, firstly, a formalism for representing a text, and therefore for representing possible specifications for anaphors; and, secondly, a set of procedures for arriving at the (hopefully) correct interpretation of a pronoun largely by inferential methods. In this chapter, I will describe the first aspect of Wilks' work; a fairly detailed description is necessary to provide sufficient background for the description of SPAR's meaning representation in the next chapter. The second part of Wilks' theory will be discussed later, in chapter 7.

Both parts of Wilks' preference semantics theory were embodied in an experimental system which I will call PS. PS translated short non-technical English texts into French, concentrating its efforts on resolving lexical, structural and anaphoric (mainly pronominal) ambiguities. The preference semantics theory may, I believe, be characterised by the following two principles.

(1) A possible interpretation of a piece of language should be judged acceptable or unacceptable not on intrinsic grounds alone, but by comparing it with alternative interpretations (if any) of the same piece of language.

(2) The basis for comparing a set of interpretations is their respective degrees of redundancy. More redundant interpretations - that is, ones that introduce less new information (in some intuitive sense) into the system - are to be preferred, since redundancy occurs when parts of an interpretation reinforce other parts.

These principles were applied by PS in its efforts to construct a single connected representation for a text. The objects and processes involved in these efforts are described in this chapter. Attention is focussed on the aspects of PS most relevant to SPAR; for a more complete description, see e.g. Wilks [1976a]. The chapter also includes a discussion of some of Wilks' later, unimplemented plans for extending the system and a brief description of Boguraev's analyser, which, as we have seen, provides the input for SPAR, and uses and develops some of the ideas of preference semantics. No assessment of Wilks' or Boguraev's representations is attempted in this chapter; instead, their suitability as a basis for anaphor resolution is discussed in chapter 4 as part of the motivation for SPAR's own representation.

PS underwent considerable development between the late sixties and mid seventies. The version of the system to be discussed here is, as far as possible, the final one, as described in Wilks [1975a, 1975b, 1975c, 1976, 1977a, 1977b, 1978].

3.1 Semantic primitives and formulas

Each open-class word in the dictionary used by PS was assigned one or more senses, each defined, in as much detail as desired, by a <u>semantic formula</u> represented as a tree of primitive elements. Formulas were intended, according to Wilks, to contain information about the meaning of the word rather than facts about the thing it stood for. The set of primitives, and the rules for constructing formulas from them, evolved over the years, and no strong claims were made for their intrinsic correctness; their justification was purely in terms of the overall system performance.

In the fullest description of the system of formulas and primitives Wilks [1977a], just under a hundred primitives were defined, most of them being categorised as substantive (e.g. MAN, STUFF, THING), case (SUBJ, INST), action (MOVE, CAUSE, WANT) or qualifier (TRUE, GOOD) primitives. In addition, a number of <u>class primitives</u>, whose names began with asterisks, were defined to correspond to a set of ordinary ones; so for example *HUM covered both MAN and FOLK. Any primitive could in principle be prefixed by NOT to negate its meaning.

A formula consisted of at least a head primitive, broadly categorising the word sense,[1] and, for verb senses, subtrees expressing preferential constraints on role fillers. So for example, the formula for the verb "interrogate" was

(3-1) `((MAN SUBJ) ((MAN OBJE) (TELL FORCE)));`

this defines "interrogate" as "force to tell something", done preferably by persons ("MAN") to persons. Each bracketed subtree in a formula consists of a <u>governor</u> (primitive or subtree) on the right, and a <u>dependent</u> (primitive or subtree) on the left; the ultimate governor, or head element, of the entire formula therefore always occurs at the extreme right, with role names occurring to the right of constraints on their fillers, as in (*HUM SUBJ).

As well as a head and some role constraints, formulas could contain qualifier elements and, for noun senses, what Wilks called an <u>inverted nominal</u> construction; for example, "policeman" might be defined as

(3-2) `(((FOLK SOUR) (((NOTGOOD MAN) OBJE) PICK)) (SUBJ MAN))`

where the case primitive SUBJ, occurring as dependent of the rightmost MAN, specifies that the MAN is the agent of the action described by the subtree headed by PICK . In (pseudo-)English, this formula defines a policeman as "a person who selects bad persons out of the body of people (FOLK)".

---

[1]In fact this function could also be performed by a "conventional subformula" such as (THRU PART), denoting an aperture, whose meaning was more than the sum of its parts and which was largely treated by the system as an irreducible object.

Wilks intended formulas to be "habitable" - that is, easy to read and write. They therefore tended to have considerable implicit content; in the formula for "interrogate" above, the fact that the two MAN primitives represent different entities must be determined using the rules for formula interpretation, as must the participant(s) in the TELL action. What is forced is not just a generalised "telling" (as represented by the TELL primitive, which has no dependents, on its own) but a telling of something by the object of the interrogation.

3.2 Fragmentation and template matching

PS contained no explicit syntactic knowledge. The first phase of its processing was a fragmentation procedure which divided the text into small strings of a handful of words each, according to the occurrence of certain predefined key words, typically function words. The heads of the formulas for the words in each fragment were matched with bare templates taken from an inventory of several hundred. A bare template, such as [MAN FORCE MAN], consisted of an ordered triple of primitives representing the agent, action (in a very broad sense) and object (the first and last of which might be dummies) of what Wilks called a "basic message". A successful match resulted in a full template (henceforth, simply template) being constructed, in which each of the primitives in the triple was replaced by the corresponding formula. In general there would be several templates for each fragment.

As an example (adapted from Wilks [1975a]), consider the sentence "The policeman interrogated the crook / at night" (fragmented at the "/"), where "policeman" and "interrogate" have the definitions given above, and "crook" has two definitions,

(3-3a) (((NOTGOOD ACT) OBJE) DO) (SUBJ MAN))
(3-3b) (((THIS BEAST) OBJE) FORCE) (INST (LINE THING)))

for the "criminal" and "shepherd's stick" senses respectively. For the first fragment, the bare templates [MAN FORCE MAN] and [MAN FORCE THING] will match the formulas for the words "policeman interrogate crook" in the first fragment (one for each sense of "crook"), to yield a full template

```
(3-4)   [   ((FOLK SOUR) ((NOTGOOD MAN) OBJE) PICK)) (SUBJ MAN))
            ((MAN SUBJ) ((MAN OBJE) (TELL FORCE))
            (((NOTGOOD ACT) OBJE) DO) (SUBJ MAN))     ]
```

and a second one with the "shepherd's stick" formula in third position. From now on, I will follow Wilks' convention of indicating templates by the words they correspond to; thus the two full templates derived from our first fragment will be written

(3-5a) [the+policeman interrogated the+crook(MAN)]

and

(3-5b) [the+policeman interrogated the+crook(THING)]

though it must be remembered that their slots are actually filled by formulas and not by words.

The second fragment, "at night", where "night" might have the formula (THIS (WHEN SPREAD)) ("a specific period of time"), would match with a bare template like [Ø DBE *ANY], where Ø is a dummy and DBE matches prepositions, to give a full template

(3-6) [Ø at night].

Next came the first significant application of the two principles of preference semantics given earlier. The SUBJ and OBJE preferences of the action formula in each template (if any) were examined and compared with the formulas filling the agent and object slots respectively. The competing template(s) with the smallest number of unsatisfied preferences were retained. Thus in our example, template (3-5a) contains two satisfied preferences (the agent and object slots of the template are both filled by formulas whose head is MAN, as preferred by the "interrogate" formula), but in (3-5b) only one preference is satisfied, since the second formula for "crook", whose head is THING, does not satisfy the object preference for MAN. (3-5b) is therefore rejected. However, if the sentence had contained the word "stick" instead of "crook", and if "stick" only had one sense, defined with a formula such as (THIS (LINE THING)), the resulting template would not be rejected in spite of its broken preference, because of the lack of any better alternative.

PS then attempted to link pairs of templates together by applying paraplates, which were structures representing the uses of prepositions and other function words. They expressed constraints on the contents of slots in the two templates they were intended to link. A constraint could apply to a single slot (such as specifying what the head of a formula should be) or to a relationship between several slots (such as specifying that they should contain the same subformula in a certain position). A successful paraplate match resulted in the establishment of a case tie between two particular slots in the two templates. Thus for our example, a paraplate stored under the preposition "at" might specify that if the first template had an action-slot formula whose head matched *DO (a very general action primitive), and the second had an object-slot formula whose head was the conventional subformula (WHEN SPREAD), then the templates could be linked by a case tie labelled TLOC (time location). This would give the representation (from (3-5a) and (3-6))

```
(3-7)      [ the+policeman    interrogated    the+crook(MAN) ]
                                   ↑
                                  TLOC
                                   |
                             [ Ø   at    night ]
```

After paraplates were applied, PS would go on to resolve any pronouns in the representation and then to generate a French translation. The way this was done is described in later chapters in the context of other, related work.

Wilks' system of semantic primitives and formulas plays an important role in SPAR, as does the process, illustrated in this section, of disambiguation by preference maximization. However, the use of these ideas in SPAR is in a sense mediated by their use in Boguraev's [1979] analyser, from which SPAR takes its input; and indeed part of SPAR's semantics derives from ideas original to Boguraev. It is therefore appropriate at this stage to consider Boguraev's work.

3.3 Boguraev's analyser

Boguraev's analyser (henceforth BA) has access to two sources of linguistic knowledge: a grammar of English encoded as an augmented transition network (ATN; Woods [1972]), and word-specific information in a dictionary. The central part of each word sense definition is provided by a Wilksian word-sense formula.

The processing performed by BA, as opposed to the structures it produces, is not of direct importance to SPAR and will be discussed only briefly.

At various points during ATN-driven syntactic parsing, routines are called to construct, using semantic information from the dictionary, one or more semantic structures for the constituent just recognized. If no coherent structures can be produced, the parser backtracks, thus avoiding following up incorrect paths. This extra efficiency is achieved at the price of making BA to a certain extent a filtering system rather than a preference one; if the only interpretations possible for a sentence contain broken preferences, PS would still produce an interpretation, whereas BA will sometimes (depending on the syntactic relationship associated with the broken preference) not do so.

The sentence representations produced by BA are dependency trees with case-labelled components; for full details see Sparck Jones [1984]. The representations differ from the structures produced by PS in reflecting syntactic structure more explicitly, and in being verb-, rather than template-, centred. Thus while PS represented the agent and object of an action differently from other participants, by direct inclusion in the template rather than by means of a case tie, BA represents all role fillers in the same way. Thus for the sentence "The policeman interrogated the crook at night", assigned the representation (3-7) by PS, BA produces

```
(3-8)   (CLAUSE (TYPE DCL) (TNS PAST)
          (V
            (INTERROGATE1 FORCE
              (@@ AGENT (N (POLICEMAN MAN (@@ DET (THE1 ONE)))))
              (@@ TIME-LOCATION (N (NIGHT1 SPREAD)))
              (@@ RECIPIENT
                (N (CROOK1 MAN (@@ DET (THE1 ONE)))))))))
```

where the structures originating from clauses and noun-phrases are marked by "CLAUSE" and "N" respectively, and all case roles (marked by @@) are represented uniformly.[2]

Relative clauses and adjectival modifiers are represented by BA with a "trace" mechanism as follows. For the sentence "The policeman arrested the man who

---

[2]In more detail, this dependency structure may be read as follows. The input sentence is a clause of declarative type and past tense, whose main verb has the sense "INTERROGATE1"; the head primitive of the Wilksian formula for INTERROGATE1 is FORCE. The explicitly mentioned role fillers of the interrogating action are the policeman, the crook and the night, each expressed by noun phrases; the first two both having the determiner "the" (modifiers such as determiners being represented like more conventional Fillmorean case roles).

stole the red car", BA produces the structure in figure 3.1. Each of the two "TRACE" structures in this representation contains a pointer to the head of the relative clause it expresses. The bottom clause, with verb sense BE2, would (apart from syntactic features) be the same as those generated for "The car was red" and "The colour of the car was red".

```
(CLAUSE (TYPE DCL) (TNS PAST)
   (V
      (ARREST1 PICK
         (@@ AGENT (N (POLICEMAN MAN (@@ DET (THE1 ONE)))))
         (@@ RECIPIENT
            ((TRACE (CLAUSE V AGENT))
               (CLAUSE (TYPE RELATIVE) (TNS PAST)
                  (V
                     (STEAL1 SENSE
                        (@@ AGENT (N (MAN1 MAN (@@ DET (THE1 ONE)))))
                        (@@ OBJECT
                           ((TRACE (CLAUSE V AGENT))
                              (CLAUSE
                                 (V
                                    (BE2 BE
                                       (@@ AGENT
                                          (N
                                             (CAR1 THING
                                                (@@ DET (THE1 ONE)))))
                                       (@@ STATE
                                          (ST (N (COLOUR NIL))
                                             (VAL
                                                (RED1
                                                   KIND)))))))))))))))))))
```

Figure 3.1: BA's analysis of
"The policeman arrested the man who stole the red car"

Another important difference between BA and PS is that BA delivers semantic interpretations of sentences, whereas PS did not attach any particular importance to sentence boundaries, at least as compared to clause (fragment) boundaries. (This is related to the fact that BA, unlike PS, contains an explicit syntactic component). This sentence-orientation, together with the fact that BA ignores anaphoric ambiguity, has important consequences for SPAR's processing strategy.

The case role names used by BA (AGENT, LOCATION etc.) are different from, and rather more discriminating than, the case primitives used by PS. However, the formulas in BA's dictionary contain Wilks' PS case primitives, and there is no straightforward mapping between the two sets. SPAR converts the BA role names into Wilksian primitives by a "hacked" method when constructing its text representation, since the whole problem is a historical artifact rather than a genuine linguistic phenomenon.

3.4 Unimplemented extensions to the PS framework

In two papers written after development of PS ceased, Wilks proposed two important extensions to the system, one concerning a modification to the inference mechanism, and the other enabling the system to deal with "extended use of language", i.e. metaphor. The first extension is incorporated into SPAR, as described in chapter 8; the second, described here, is not, for the reasons given below.

In Wilks [1978], a scheme for actively re-interpreting broken preferences, especially those embodied in metaphor, using script-like objects called pseudo-texts (PTs) was put forward. A template containing a broken preference, such as that for "My car drinks petrol", where the formula for "drink" specifies a preference for an animate agent, would be compared with templates in the PT(s) associated with objects in that template. The closest match would be used to re-interpret the template with the broken preference. In our example, if the PT for "car" contained a template stating that its engine used a liquid, then "drinks" would be interpreted as meaning "uses". Wilks pointed out that the idiomatic element of this use of "drinks", i.e. that it means "uses a lot of", could not even in principle be inferred, and would have to be pre-stored explicitly.

Pseudo-texts are not implemented in SPAR because they appear to be less relevant to anaphor resolution than to other aspects of interpretation, and because, when faced with a situation in which only interpretations involving broken preferences can be constructed, Boguraev's analyser often (although in fact not for the case "my car drinks petrol") assumes that a wrong syntactic path has been followed, and backtracks without returning any interpretation for the current path. Such behaviour is essential for reasonably efficient processing. Thus if pseudo-texts were to be used, they would have to be applied during parsing on any occasion that backtracking was considered. But as pseudo-texts are fairly complex objects, this would make the analyser far too inefficient.

More recently, Fass and Wilks [1983] have outlined an alternative to pseudo-texts. They propose to split the metaphor interpretation process into a relatively cheap and simple detection phase, which merely recognises that a metaphor is occurring, followed by a more complex interpretation phase, which determines the metaphor's meaning. Although Fass and Wilks do not discuss the issue, it would seem that only the first phase would be necessary for an analyser such as Boguraev's to decide whether or not to backtrack. However, the assumption that metaphor detection can in fact be separated from metaphor interpretation in the way proposed by Fass and Wilks is a questionable one, as argued in Carter [1984].

Thus neither the pseudo-text idea nor Fass and Wilks' later proposals are embodied in SPAR, because metaphor interpretation is a rather different problem from anaphor resolution, and because of the difficulty of invoking the appropriate processes efficiently and at the right moment.

4. Meaning Representation in SPAR

"Nothing can please many, and please long, but just representations of general nature."

Samuel Johnson, "Preface to Shakespeare".

This chapter describes the way that SPAR represents both the stories it processes and the pre-stored linguistic knowledge which plays a central role in that processing. Because SPAR takes input from Boguraev's analyser, the representational formalism is quite strongly based on that of Boguraev. As we saw in the last chapter, Boguraev's representation is in turn partly based on Wilks', specifically in its use of word sense formulas and more generally in its overall philosophy. Rather than presenting an independent and general justification of SPAR's representation, therefore, I will assume that what one might call the "Wilks/Boguraev approach" to representational issues is broadly appropriate for performing language-oriented tasks such as translation or paraphrase, and will build on that approach.

In section 4.1 I will argue that a Wilks/Boguraev approach demands that, as far as possible, structures of the same general character should be used to represent the text at all stages of processing, and that therefore SPAR's representation should resemble Boguraev's as far as possible. However, the fact that SPAR is oriented towards resolving anaphoric ambiguities in texts, rather than, like Boguraev's analyser, word-sense and structural ambiguities in single sentences, means that certain differences are necessary. I will outline the general nature of these differences and give an overview of the rest of the chapter, which presents SPAR's representation in detail.

Section 4.2 describes in detail SPAR's representation of pre-stored linguistic knowledge, while section 4.3 discusses the representation of texts. The form of representation as a whole is summarised and evaluated in section 4.4.

At various points in this chapter I will comment on SPAR's representation scheme compared with the two knowledge representation languages KRL (Bobrow and Winograd [1976]) and KL-ONE (Schmolze and Brachman [1982]).

4.1 Requirements

4.1.1 Design Methodology

We have seen that in Wilks' PS system a text was represented at almost every stage of processing by a set of interlinked actor-action-object templates. Templates corresponding to input phrases were constructed early on in the analysis phase, and templates continued to represent the input throughout subsequent operations, including pronoun resolution, until they were finally used as the input to a French generator. Although the representation was deepened somewhat during processing, it remained template-based.

This policy of working throughout with a representation quite closely related to the input surface text is especially appropriate for a system which performs a language-oriented task such as paraphrase or translation rather than an event-oriented one such as question answering, since language-oriented tasks seem on average to demand less inference. If the representation is made more regular and canonical, for example by removing traces of surface words and structures and filling in missing case roles, it will tend to become more amenable to inference, but the tasks of constructing such a representation and generating natural language from it become correspondingly more difficult. The representation must be designed with this trade-off in mind; if, as in our shallow processing approach, comparatively little inference is to be performed, then a relatively superficial representation (i.e. one whose form reflects that of the input text quite closely) is appropriate.

SPAR takes Boguraev's relatively superficial dependency structures as input, and, as chapter 9 will describe, hands such structures to a modified version of Tait's English generator as output. It also exploits the Wilksian word sense formulas used by Boguraev's analyser. The more SPAR's internal representation differs from Boguraev's, the more difficult will be the task of transforming one into the other. Similarly, other things being equal, we will want to transform the formulas in the dictionary as little as possible. The principle used for designing SPAR's meaning representation is therefore the following:

> SPAR's representation of texts should be based as closely as possible on Boguraev's dependency structures, diverging from them only where the differences between the orientations of the two systems require it. Similarly, SPAR's representation of pre-stored linguistic knowledge should diverge from the word sense formulas in the analyser dictionary only where the differences in system orientation require it. The two main such differences are: (1) SPAR is oriented mainly towards resolving anaphoric ambiguities, while Boguraev's system considers only word-sense and structural ambiguities. (2) SPAR works with multi-sentence texts, while Boguraev's system analyses single sentences in isolation.

We will now look at the main way in which Boguraev's representation is unsuitable for anaphor resolution in connected texts, and will argue that, despite its use for anaphor resolution, Wilks' template-centred representation was inadequate in just the same way.

4.1.2 The need for a text model

The main drawback of both Boguraev's and Wilks' representations from the point of view of anaphor resolution can be summarised by saying that they make explicit only information about the *senses* of the words, phrases and clauses they process and leave information relevant to their *specifications* implicit. That is, the representations are both sense-oriented; one might even say that the entities in PS's semantic blocks corresponded only to (particular senses of) words in the text and not in any direct way to the objects and events the text described.[1]

An important consequence of this is that neither representation makes explicit provision for distinguishing between distinct entities sharing the same sense. Wilks [1975b, p73] comments, about PS's treatment of the sentence

(4-1) John drank the whisky from a glass and it felt warm in his stomach.

"...it is not specified in the notation [i.e. in the representation] whether the whisky was, or was not, the Winogradian :WHISKY, which is to say was it or was it not particular whisky, different from other samples of whisky. This is a distinction which makes most sense within a micro-world of inventoried items and samples, and less so outside."

In practice the lack of such a distinction was not a problem for PS because of its task (English to French translation) and its text type (short, and/or without significant anaphora problems). For example, when processing

(4-2) John traded in his old car for a new one. It broke down within a week.

it is not necessary to decide which car broke down, or even to realise that more than one car is mentioned, in order to translate "it" by a French pronoun of the correct gender. Either way, "it" refers to a car, which determines the gender.

However, for longer texts, or for other tasks requiring more detailed reasoning, a sense-oriented approach is unlikely to be adequate. If (4-2) occurs in the middle of a long story which first talks about John's old car and then, after the trade in, talks about the new one, then any inference necessary to choose between the (new) car and other candidate pronoun referents in the later part of the story must avoid mistakenly accessing information about John's old car.

Furthermore, a focus mechanism such as Sidner's will be misled if similar entities are not distinguished. A second reference to a recently-mentioned entity is likely to make that entity strongly focussed, whereas the introduction of a new entity similar in description to an earlier one will not make either of them so strongly focussed. In fact, Sidner assumes, and shows the need for, a representation in which specific instances are distinguished both from one another and from generic concepts.

In addition, in chapter 7 we will see that an inference mechanism designed along Wilksian lines should be sensitive to intensional context; it should only make inferences on the basis of what is actually asserted in the text. The representation on which inference is performed should therefore also express intensional context.

We may conclude that Wilks' use of a simpler, sense-oriented representation was adequate for the limited number of short texts on which Wilks

---

[1] This is perhaps suggested by Wilks' [1978,p210] belief that one should "stress the form of representation of language and seek to accommodate the representation of knowledge to that, rather than the reverse."

demonstrated PS's pronoun resolution mechanism, but that a system with more ambitious aims would have to represent thoroughly not just the semantic patterns in the text but also the nature and identity of the objects and situations described, as determined by factors like intensional context, quantifier scope and genericity (see e.g. section 2.3). Thus some alteration to Boguraev's representation, which is similarly sense-oriented, is also needed. More specifically, our representation must qualify as a text model of the type assumed in the characterisation of anaphora and specification given in section 2.1. That is, there must be at least approximately a one-to-one correspondence between model elements and story entities (in a very broad sense of the latter). We will see later in this chapter that this is true of SPAR's representation.

In 4.1.1 above we noted that SPAR's orientation differs from that of Boguraev's analyser in two main respects: in concentrating on anaphora, and in processing connected texts rather than isolated sentences. The drawbacks for our purposes of Boguraev's representation discussed until now have all been consequences of the first difference. However, the second difference also has important consequences, in that Boguraev's structures are not of a tractable form for representing texts as opposed to sentences. Their hierarchical form is appropriate for a sentence representation, because sentences exhibit governor/dependent relationships such as those of main clause to subordinate clause and noun phrase to dependent relative clause, and these relationships are arguably part of the meaning of the sentence. However, a hierarchy is less sensible for a text representation, in which there is usually no single obvious "main clause", and the relationships between sentences or propositions are not necessarily best thought of as being of the governor/dependent type.

A related point is that because Boguraev's analyser does not attempt to identify cospecifying noun phrases in a sentence, there is usually a one-to-one correspondence between noun phrases in the sentence and the "noun-args" that represent them in the dependency structure. This allows the structure to be tree-like. However when, in a system such as SPAR, the decision is made that two noun phrases cospecify, it is desirable, for the sake of simplicity, to merge the structures representing them. Merging makes the hierarchy (if there is one) tangled and more difficult to work with. Furthermore, SPAR often needs to know what has been said about a given entity, or whether a particular pattern of elements and case links occurs in the text representation, and such information is far more easily recovered from a network than from a hierarchy, whether tangled or tree-like. Neither of these arguments absolutely rules out the use of a hierarchical representation, but they do mean that the fairly minor task of transforming Boguraev's hierarchical structures into heterarchical (network) form is worthwhile.

4.1.3 Overview of SPAR's meaning representation

Earlier in this chapter we established the principle that SPAR's meaning representation should differ from Boguraev's as little as is consonant with the needs of anaphor resolution and of processing texts as opposed to sentences. We have also seen that the form of representation should be heterarchical and not hierarchical, and that it should qualify as a text model (that is, that elements in the representation should approximately correspond one-to-one to story entities). The shallow processing approach dictates that we should aim to make as much use of linguistic knowledge as possible. On the basis of the discussion of lexical cohesion in subsection 2.2.3, it would seem that the representation must make it possible to determine not only whether an anaphor and suggested cospecifier are semantically compatible, but also whether the preferential "information constraint" (that the anaphor does not introduce new information) is satisfied, and whether collocation is a possibility. Thus in

(4-3) [1] John has bought a new Jaguar.
[2] The car is ultra-modern in every respect.
[3] For example, the indicators use the latest laser technology.

the information that Jaguars are types of cars and typically have indicators must somehow be retrievable for the underlined anaphors in [2] and [3] to be resolved.

In SPAR it is assumed, as a corollary of the shallow processing hypothesis, that the necessary information can normally be extracted from the word sense formulas for the words involved. Formulas encode linguistic knowledge that is essential to a basic understanding of the word senses they define. Examples of such knowledge are that a "Jaguar" is a type of car, and that an "indicator" (in the appropriate sense) is part of a vehicle.[2]

Thus the process of resolving "the car" above must involve reasoning that a Jaguar is a car, so as to verify that "a new Jaguar" and "the car" can cospecify without "the car" introducing any new information. To resolve "the indicators", the fact that Jaguars have indicators need not (and probably will not) be stored explicitly as part of the definition of either "Jaguar" or "indicator". However, that fact can be deduced from the knowledge that Jaguars are cars (in the formula for "Jaguar"), which are road vehicles (in the formula for "car"), and indicators are parts of road vehicles (in the formula for "indicator"). "The indicators" can therefore be understood anaphorically as "the indicators of John's new Jaguar".

In SPAR, the information in each formula is transformed into a set of assertions in the formalism of Alshawi [1983]. Assertions relate the sense being defined both to primitives occurring in other formulas and directly to other senses; thus the assertions for all the senses involved in a story together make up a word sense network (WSN).

When a noun phrase or clause is encountered in the input, provisional text model elements are created not only for the specification of the phrase or clause itself but also for any other entities mentioned in the formula for its head word sense (as represented in the WSN). Thus if the formula for an indicator defines it as a particular part of a road vehicle, then when "the indicators" is first encountered, elements, possibly temporary, for both the

[2]As noted in chapter 1, there is ultimately no principled way to decide whether a given fact about a concept counts as basic definitional knowledge which should be included in a formula, or as world knowledge which should not. The fact that a Jaguar is a car is clearly definitional; the fact that indicators are parts of vehicles is less obviously definitional but is still arguably one of the handful of most central facts about indicators.

indicators and an unidentified road vehicle, are copied from the WSN to the text model network (TMN). The first of these elements is explicit (because it represents something explicitly mentioned in the text); the second is implicit.

Collocation can then be treated as a generalisation of reiteration; during anaphor resolution, not only are pairs of explicit elements matched, but also, as Sidner's noun phrase resolution algorithm (to be discussed in chapter 5) in fact demands, pairs composed of an explicit and an implicit one. In [3] in our example, the explicit elements for the indicators and the Jaguar fail to match, but the implicit element for the indicators' unidentified road vehicle and the explicit one for the Jaguar do match.[3]

This process of copying pieces of network from the WSN to the TMN makes the representation of explicitly-mentioned entities uniform with that of entities not mentioned but implied to exist. This leads not only to the uniform treatment of reiteration and collocation but also, as we will see later, to some advantages in common sense inference. The TMN is not made unmanageably large because formulas, and hence the WSN, contain only the most basic information about word meanings, and so the number of implicit elements per explicit one is small. This would not be the case if the mention of an object (e.g. a car) caused elements for all possible subsequent collocative anaphors (the wheels, the driver, the mudguards...) to be constructed. Thus the practice of including only basic definitional knowledge in formulas leads ultimately to a representation which is detailed enough to allow most collocative anaphors to be resolved, but simple enough to allow uniformity between explicit and implicit elements.

In the next section, 4.2, the WSN and its derivation from formulas is described in detail. Section 4.3 then covers the construction of new pieces of TMN from the incoming dependency structures and relevant parts of the WSN, and the TMN's use in determining the semantic acceptability of suggested antecedents.

4.2 The representation of word senses

As we saw in chapter 3, Wilks' word sense formulas provide a convenient and quite flexible way of characterizing word meanings. However, this flexibility is accompanied by considerable vagueness, which has three main causes:

---

[3] Pairs of implicit elements are not matched, because this tends to lead to too distant and unreliable a link between anaphor and (supposed) antecedent. However, I am grateful to Graeme Ritchie for pointing out that such matching can be necessary; an example is

(4-4) I must phone the garage. The engine's misfiring and the indicators are on the blink.

However, such examples seem to be unusual and slightly strained in a neutral context; (4-4) would be most natural in a context where the speaker's car was well known to both participants, and therefore as salient as if it had been explicitly mentioned. The underlined anaphors could then be resolved using explicit-implicit pairings.

(1) As Wilks acknowledged, the conciseness and expressive power of formulas in PS was severely limited by their only containing primitives and not word senses.

(2) Wilks' efforts to make formulas "habitable" - easy to read and write - led him to allow certain relationships within them to be present only implicitly; to make them explicit, the interpretation rules in Wilks [1977a] had to be applied. However, because habitability made formulas look like statements in a kind of sub-English, the writers of dictionary entries for Boguraev's system have tended to treat them as such, without regard to the rules. (This may also be true of Wilks; however, since the primitives and formulas evolved over time, it is difficult to be sure). Formulas have therefore often inherited some of the vagueness and ambiguity of natural language. Although habitability and implicit relationships are in theory compatible with correctness, in practice they tend to reduce it. Neither Boguraev's system nor, judging by published descriptions, Wilks', in fact used Wilks' [1977a] interpretation rules directly; Boguraev's system normally accesses only the head primitive and top-level cases in a formula, while Wilks' relied on "fuzzy matching" between formula subparts which would have been largely unaffected by any syntactic errors. However, if the information in formulas is to be fully exploited, a more rigorous approach to ensuring its accuracy is needed.

(3) Although the syntax of formulas is formally defined in Wilks [1977a], the semantics accompanying that syntax is specified only informally. One consequence is that there is not always a principled way to decide whether the formulas for two word senses match (i.e. whether the word senses could correctly be used to describe the same entity), and if so, whether one contains information that the other does not.

These defects are remedied in SPAR, for (1) by allowing word senses in formulas; for (2) by defining an intermediate formula syntax in which implicit information is made explicit; and for (3) by automatically converting each formula via the intermediate form into a set of relations of the type used by Alshawi [1983], whose semantics are well enough defined for our purposes and which allow easier matching. These three modifications are described in the next three subsections.

4.2.1 Word senses in formulas

Wilks [1976, p169] believed the chief drawback of his system to be "that codings consisting entirely of primitives have a considerable amount of vagueness and redundancy. For example, no reasonable coding in terms of structured primitives could be expected to distinguish, say, "hammer" and "mallet". That may not matter provided the codings can distinguish importantly different senses of words. Again, a template for the sentence "The shepherd tended his flock" would contain considerable repetition, each node of the template trying, as it were, to tell the whole story by itself. Whether or not such a system can remain stable with a considerable vocabulary, of say several thousand words, has yet to be tested." This problem can at least partially be solved, as Wilks [1978] pointed out, by allowing formulas to contain word senses as well as primitives, as long as those word senses are themselves defined elsewhere. SPAR implements this idea, which confers two main advantages.

Firstly, formulas become more concise and less redundant. "Food" is best defined in terms of eating; if we define (one sense of) "eat" by

```
EAT1:      ((*ANI SUBJ) ((STUFF OBJE) (((SELF IN) MOVE) CAUSE)))
```

("an ANImate being CAUSEs some STUFF to MOVE INto him/herSELF") then "food" can be defined by

```
FOOD1:     (EAT1 (OBJE STUFF))
```

("STUFF which is EATen"). This can be (and is, for use by Boguraev's analyser) expanded to a primitive-only formula by substituting EAT1's formula for EAT1 and identifying the two occurrences of STUFF. However, the unexpanded version is easier to read and write, and therefore more habitable. It also eliminates redundancy and therefore the risk of error; a change in the formula for EAT1 will not prevent a pattern matcher from seeing the similarity between the formulas, as it might if FOOD1 were defined without explicit reference to EAT1. There is little risk that any reasonable change in EAT1's formula will render that for FOOD1 incoherent, because however we define EAT1, it is hard to imagine that FOOD1 could be related to it in any way other than being its OBJEct.

Secondly, formulas can be given greater discriminatory power if an additional device is introduced. In SPAR, a (sub)formula of the form (THIS X) is used to mean "a distinct type of X, different in an unspecified way from all other distinct types of X".[4] Thus different species of animal can conveniently be defined as follows:

```
SHEEP1:    (THIS BEAST)
HORSE1:    (THIS BEAST)
ELEPHANT1: (THIS BEAST)
```

SPAR can therefore deduce that the phrases "the sheep" and "the horse" can never cospecify, even if it has no knowledge of *how* the two species differ. Not only does the THIS mechanism provide a succinct and usually sufficient definition; it is also theoretically appropriate, since the nature of the difference between a sheep and a horse is perhaps more naturally categorised as world knowledge than linguistic knowledge. (The THIS mechanism also has the effect of making it impossible even in principle to determine senses from formulas. Senses therefore play a crucial role in processing in their own right, and in particular are the basis on which paraphrases are selected).

The use of THIS to indicate an incomplete definition has some similarities to KL-ONE's use of "starred concepts" to represent incompletely defined "natural kinds". However, the main use of THIS is to mark pairs of word senses (such as SHEEP1 and HORSE1) as disjoint; analogous pairs of starred concepts in KL-ONE are not necessarily disjoint.

---

[4]Wilks [1977a] defines THIS as a substantive primitive meaning "an unidentified, but particular, entity of any type". This may or may not be equivalent to my use.

Formulas for word senses related to those defined using THIS can be made quite discriminating. If ELEPHANT1 is defined as above, we can define

```
TRUNK1:     ((ELEPHANT1 POSS) NOSE1)
```

("the nose of an elephant"), assuming that NOSE1 has its own formula. A primitive-only formula could not express the fact that it is elephants, and not sheep or horses, that have trunks, because ELEPHANT1 has the same formula as SHEEP1 and HORSE1. Such facts can be important in resolving anaphors:

(4-5) The elephant approached the sheep. It waved <u>its trunk</u>.

It is sometimes useful to be able to define two word senses in terms of each other; for example, to define EAT1 in terms of FOOD1 as well as the reverse. Such relationships can be represented in formulas as long as there is no circularity in head position. If we replace STUFF in the formula for EAT1 by FOOD1, then the formula for EAT1 can be expanded into primitives without infinite regress, simply by replacing any embedded occurrences of EAT1 by EAT1's head primitive, CAUSE. However, if there is circularity in head position, this is not possible:

```
MONKEY1:   (THIS APE1)
APE1:      (THIS MONKEY1)
```

4.2.2 Making formulas more explicit

As we have seen, Wilks' attempt to make formulas habitable by allowing certain information to be only implicitly present leads in practice to their being vague, hard to exploit fully, and often incorrect. In SPAR, formulas are therefore automatically recast in a more explicit <u>intermediate form</u> as a preliminary to their conversion into the word sense network itself. This makes them less natural-language-like but in fact far more readable and easier to write correctly. For example, Wilks [1977a] gives the formula for "break" (as in "John broke the window") as:

```
(4-6)    BREAK1:    ((*HUM SUBJ)
                      ((*PHYSOB OBJE)
                        (((((NOTWHOLE KIND) BE) CAUSE) GOAL)
                          ((THING INST) STRIK))))
```

meaning "a HUMan agent (SUBJ), using a THING, STRIKes a PHYSical OBject with the GOAL of CAUSE-ing the physical object NOT to be WHOLE". Wilks [1977a] gives "transgroup rules" which allow one to determine that the implicit SUBJ of CAUSE is the *HUM, and that of BE is the *PHYSOB. Using these rules, SPAR transforms the above formula to

```
(4-7)    BREAK1:    (STRIK (0)
                      (SUBJ (*HUM (1)))
                      (OBJE (*PHYSOB (2)))
                      (GOAL (BE (4)
                               (SUBJ (*PHYSOB (2)))
                               (OBJE (WHOLE (5)
                                        (TYPE (NOT))))))
                      (INST (THING (6))))
```

In this intermediate form, the head of the formula is moved into first, rather than last, position, and its case role preferences are listed after it as slot-filler pairs. This makes it clear what entities are related by case primitives; in Wilks' syntax, for example, it is not obvious without reference to the interpretation rules that the *PHYSOB is the OBJE filler of STRIK rather than, say, BE or CAUSE. In intermediate form, each substantive and nominal element has a numeric index which shows clearly whether two substructures represent the same participant fulfilling two roles, or different participants. Thus the two occurrences of *PHYSOB are equivalent because their indices are equal. In Wilks' formulas, the equivalence or non-equivalence of two occurrences of the same primitive can be determined only with reference to the interpretation rules.

Numeric indices in intermediate formulas correspond to KRL constructions of the form "(the X from ThisOne)" and KL-ONE's role-value restrictions. In this respect, as in most others, SPAR's formalism is simpler than those of KRL and KL-ONE, because it is only intended to support the types of processing necessary for a shallow processing approach to anaphor resolution. In contrast, KRL and KL-ONE aim at maximising generality and expressive power.

The TYPE role name, which I have added to those provided by Wilks, serves not to relate participants but to attach a modifier to a participant. The semantics of various TYPE modifiers are discussed in the next subsection.

"Inverted nominal" constructions in Wilksian formulas (see 3.1) are represented in intermediate form using a pseudo-role-name WHICH, corresponding roughly to the KRL "(which <Predicate> <Arguments>)" construction. The Wilksian formula

```
ASSAILANT1:     (((*HUM OBJE) STRIK) (SUBJ MAN))
```

("a MAN who STRIKes a HUMan") is converted to the intermediate formula

```
ASSAILANT1:     (MAN (0)
                  (WHICH
                     (STRIK (1)
                        (SUBJ (MAN (0)))
                        (OBJE (*HUM (2)))))
```

SPAR's transformation of formulas to intermediate form also involves a slight increase in representational depth. Boguraev's analyser treats the semantic case roles SUBJ and OBJE at top level as specifying preferences on *syntactic* subjects and objects. So for example the formula for the "speak to" sense of "address" would have the listener in OBJE rather than the more natural FOR (recipient) position. SPAR makes the appropriate changes in such formulas so as to make the (intermediate form) formulas for verbs like "address" and "speak to" more uniform. It also adjusts formulas so that each entity, state or event involved in the definition corresponds to only one distinct participant in the formula; thus the CAUSE in the formula for BREAK1 is removed because it does not correspond to a separate event or state from the STRIK or the BE, but rather to the relationship between them, and is therefore subsumed by the GOAL case link.

4.2.3 Constructing the word sense network

Accurate and well-motivated matching of word senses is fundamental to reliable anaphor resolution. To achieve this, SPAR uses a variation of the structures of Alshawi's [1983] Memory network representation to store the knowledge encoded in formulas.

Memory can be viewed as a simplification of NETL (Fahlman [1979]), both in the distinctions it can represent and in the processing of which it is capable. Alshawi designed Memory to facilitate a certain set of retrieval requests useful for language interpretation operations, including anaphor resolution, in a particular task context; he did not claim that it was able to support full inference.

Alshawi used Memory to express relationships between entities of many different kinds, such as individuals, database concepts and word senses. It has a formal semantics defined in terms of the functions ref and rel. The function ref maps a memory entity to the set of objects in the world that the memory entity may describe: in Lyons' [1968] sense, its denotatum. The function rel maps a role-owner pair of memory entities to the set of pairs of objects in the world, i.e. to a relation, that the pair in memory can describe.

Memory uses two types of assertion, Specialisation: and Corresponds:. The assertion

(4-8) (Specialisation: COMPUTER of MACHINE)

has the semantics

(S4-8) ref(COMPUTER) $\subseteq$ ref(MACHINE);

that is, given these senses of "computer" and "machine", the set of objects properly called "computers" is a subset of the set of objects properly called "machines". The assertion

(4-9) (Corresponds: DATA/PROCESSING to COMPUTER as MACHINE/ACTIVITY to MACHINE)

means that when a computer is considered as a machine, data processing is its activity. More formally,

(S4-9) rel(DATA/PROCESSING, COMPUTER) $\subseteq$ rel(MACHINE/ACTIVITY, MACHINE)

Both types of assertion may be flagged in various ways, as described below.

It turns out that, with some fairly minor alterations, the Memory formalism is adequate for the sense matching operations required by SPAR for anaphor resolution. These operations are to determine:

(1) The relationship between the denotata (i.e. ref values) of two word senses: one of equality, strict inclusion either way, intersection or disjointness.

(2) When a given word sense is used to describe an entity or event, what other entities or events are implied to exist, and what their interrelationships are.

SPAR constructs the word sense network by turning each formula involved in a story into a set of Memory assertions. The WSN is the aggregate of these assertions and of hand-coded assertions that define the relationships between primitives. Primitives therefore have the same status in the WSN as ordinary word senses, and so the term "word sense" will henceforth often be used as a shorthand for "word sense or primitive".

The formula for BREAK1 given in (4-6) is converted, via its intermediate form (4-7), into the assertions in figure 4.1.

```
(Specialisation: BREAK1 of STRIK)
(Specialisation: BREAK1/SUBJ of *HUM)
(Specialisation: BREAK1/OBJE of *PHYSOB)
(Specialisation: BREAK1/GOAL of BE)
(Specialisation: BREAK1/INST of THING)
(Specialisation: BREAK1/WHOLE of WHOLE (NEG))
(Corresponds: BREAK1/SUBJ to BREAK1 as SUBJ to *DO (E1))
(Corresponds: BREAK1/OBJE to BREAK1 as OBJE to *DO (E1))
(Corresponds: BREAK1/GOAL to BREAK1 as GOAL to *DO (E1))
(Corresponds: BREAK1/INST to BREAK1 as INST to *DO (E1))
(Corresponds: BREAK1/OBJE to BREAK1/GOAL as SUBJ to *DO (E1))
(Corresponds: BREAK1/WHOLE to BREAK1/GOAL as OBJE to *DO (E1))
```

Figure 4.1: Memory assertions for the formula for BREAK1.

Each of the Specialisation: assertions in the figure defines one of the indexed items in (4-7); thus the *PHYSOB with index (2) becomes the node BREAK1/OBJE (the name itself is arbitrary as far as the program is concerned). The Corresponds: assertions define the case relationships these items participate in; for example, they say that BREAK1/OBJE is both the OBJEct of BREAK1 and the SUBJect of the event BREAK1/GOAL.[5]

The semantics that SPAR assigns to Memory assertions is slightly different from Alshawi's. SPAR assumes that a formula, and therefore the assertions derived from it, are "complete" in the sense that they either contain all the information needed to decide on a word sense's denotatum, or indicate explicitly that some information is missing. Thus the simple formula

```
MONKEY1:  APE1
```

which is translated to the single assertion

```
(Specialisation: MONKEY1 of APE1)
```

is taken to mean that MONKEY1 and APE1 are synonymous; their ref values are equal. In Alshawi's original formulation, the interpretation would merely be

---

[5]The fourth argument (after the second "to") in a Corresponds: assertion is practice always filled by a very general primitive such as *DO or DUMMY. This is because Wilks' set of cases is small and fixed, unlike the set of roles in Alshawi's system or the freely definable "slots" of KL-ONE or KRL.

Charniak [1981a] argues convincingly that no finite, universal set of semantic cases can be defined, since cases are in fact just slots, and can be introduced at places other than the top a hierarchy of verbs. Evidence for this is that certain sets of verbs use idiosyncratic cases (e.g. the "language" case, expressed in English by the preposition "in", for "speak") which are not applicable to other verbs. The Memory formalism is consistent with Charniak's argument since it allows assertions of the form

```
(Corresponds: FRENCH to SPEAK1-1 as LANGUAGE to SPEAK1)
```

which imply that only actions of the category SPEAK1 have a "language" slot. However, to incorporate Charniak's insight into Wilks' formalism is beyond the scope of this work (though potentially very worthwhile).

that ref (MONKEY1) $\subseteq$ ref (APE1).

If, however, it is the case that all monkeys are apes but not all apes are monkeys, the correct formula is

```
MONKEY1:    (THIS APE1).
```

This time, the THIS qualifier results in Alshawi's DISTINCT flag being added to the assertion:

```
(Specialisation: MONKEY1 of APE1 (DISTINCT))
```

The flag tells SPAR that a MONKEY1 is a type of APE1, distinct from all other types of APE1, and that there are other types of apes than monkeys: in other words ref (MONKEY1) $\subset$ ref (APE1).

Proper names are defined by formulas containing an INDIV qualifier, which also gives rise to a DISTINCT assertion. The formula

```
JOHN1:      (INDIV MAN)
```

is translated into

```
(Specialisation: JOHN1 of MAN)
(Specialisation: JOHN1 of INDIV (DISTINCT))
```

The second of these assertions states that JOHN1 is a DISTINCT INDIVidual, and prevents JOHN1 from matching with any other name (although matches with non-names are still possible).

SPAR does not use Alshawi's flags for Corresponds: assertions. Instead, it uses the flags E1, E2, and E12, which specify that a Corresponds: assertion is Essential to the meaning of either or both of its first two arguments. These flags survive when pieces of the WSN are used in TMN construction. They then affect the behaviour of the structure matcher during anaphor resolution, as will be explained later.

4.3 The text model network

Having outlined some of the characteristics required of the text representation used by a shallow processing anaphor resolver, and discussed how word sense information is represented in SPAR's WSN, it is now time to look at the way in which the TMN (text model network) is derived by combining word sense information with the structures delivered by Boguraev's analyser.

First, I will give an overview of the types of information captured in the TMN and the structures used to do this. This is followed by a description of the structure matcher used at various points in the program to determine whether two pieces of network may be regarded as equivalent. This matcher plays an important role in the derivation of provisional model elements from input dependency structures, which is then described.

4.3.1 Overview of the text model

Text model structures are created as a result of comparing incoming dependency structures with appropriate word sense nodes in the WSN. The results of matching determine how pieces of the WSN should be used as blueprints for constructing TMN structures. The TMN is therefore of the same general form as the WSN, and in particular Specialisation: and Corresponds: assertions play a key role in it; however, anaphor resolution and paraphrase make demands of the TMN that cannot be met elegantly by the devices that Alshawi's Memory formalism (or the variation of it used in the WSN) makes available, so the TMN requires a number of additional devices.

The overall process of TMN formation is depicted in figure 4.2. An incoming dependency structure is converted by the fragment constructor into the current fragment of the TMN, which is augmented with information from the WSN. SPAR's subsequent processing essentially involves deciding how to incorporate the current fragment part of the TMN into the already-existing context part; this most centrally involves resolving anaphors.

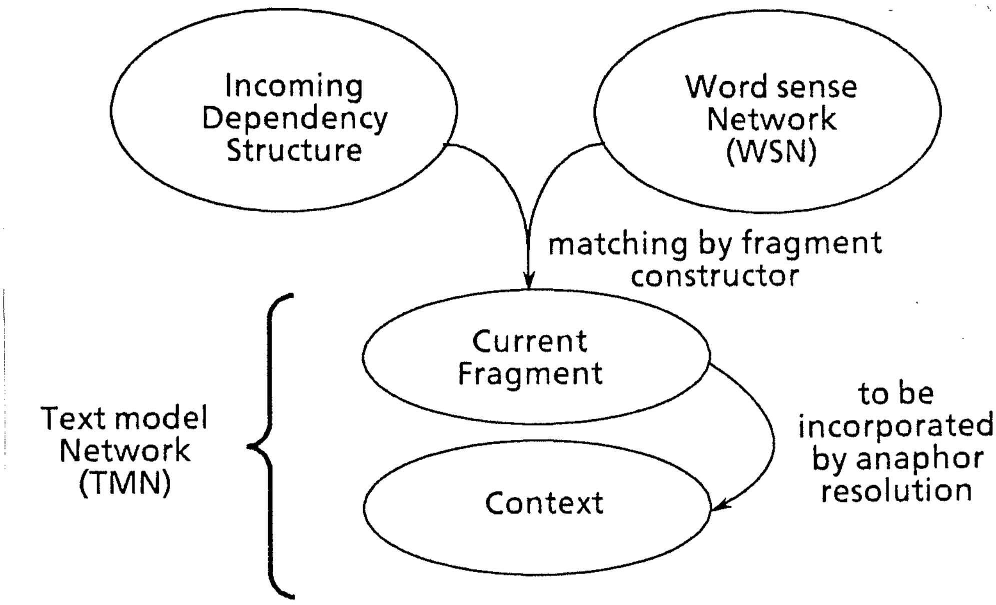


Figure 4.2: An overview of TMN formation

Just as in the WSN, TMN nodes are linked by Corresponds: assertions specifying Wilksian case relations. Specialisation: assertions relate TMN nodes to WSN nodes. However, whereas a WSN node is defined entirely by the Specialisation and Corresponds: relationships it participates in, a TMN node also has some intrinsic content, as figure 4.3 indicates. The figure shows the dependency structure for the sentence "He wanted to break a window", plus all the information associated with the TMN node representing the "break" action in that sentence at the point when the current fragment for that sentence has been formed but not incorporated by anaphor resolution into the context. (Later we will see the changes that result from incorporation into the context provided by an initial sentence "John was angry").

The various types of information exemplified by the description of BREAK1-1 in the figure, which SPAR generated automatically, are as follows. The first four types described are essentially WSN-derived; the other four are unique to TMN elements.

```
(CLAUSE (TYPE DCL) (TNS PAST)
   (V
      (WANT2 WANT (@@ AGENT (N (HE1 MAN)))
         (@@ MENTAL-OBJECT
            (CLAUSE (TNS PRESENT)
               (V
                  (BREAK1 STRIK (@@ AGENT (N (HE1 MAN)))
                     (@@ OBJECT
                        (N (WINDOW1 PART (@@ DET (A1 ONE))))))))))))
```

```
Description of TMN element with name BREAK1-1

Specialisation assertions:

   (Specialisation: BREAK1-1 of BREAK1)

   Head primitive is STRIK

Correspondence assertions:

   (Corresponds: HE1-1 to BREAK1-1 as SUBJ to DUMMY (E1))
   (Corresponds: WINDOW1-1 to BREAK1-1 as OBJE to DUMMY (E1))
   (Corresponds: BREAK1/GOAL-1 to BREAK1-1 as GOAL to DUMMY (E1))
   (Corresponds: *INST-1 to BREAK1-1 as INST to DUMMY (E1))

   (Corresponds: BREAK1-1 to WANT2-1 as OBJE to DUMMY (E1))

Status is:              Specific and new

Context of identity is: desires of HE1-1

Derivation:             directly from the BREAK in
                        HE WANTED TO BREAK A WINDOW

Linguistic features are:

   CATEGORY        =    CLAUSE
   RECENCY         =    4
   TNS             =    PRESENT
```

Figure 4.3: Dependency structure for "He wanted to break a window", and resulting TMN element for "break".

(1) The name of a node is essentially arbitrary; it is used merely as a handle to distinguish the node from all others, and for diagnostic purposes.

(2) A Specialisation: assertion is associated with each TMN element, linking it to the WSN element (word sense or primitive) from which it was derived. When one TMN element is merged into another, the Specialisation: assertions for the

first are, unless redundant, transferred to the second.

(3) The primitive attached to the node is the most specific nominal or action primitive superordinate to all the node's word senses given by (2). Its main purpose is to allow more efficient TMN node matching.

(4) The `Corresponds:` assertions indicate case relationships between nodes, and are derived from analogous assertions in the WSN and/or in the dependency structure.

(5) The status of a node is one of the values "null", "generic" or "specific":

> (a) Null nodes are rare; they prevent cospecifications being sought for phrases such as the "it" in "It was raining".
>
> (b) A generic node, as the name implies, represents a class of entities or events, and can only cospecify (trivially) with a generic node for the same class. The "banana" node in "John liked bananas" would be generic.
>
> (c) A specific node represents a particular event, entity or determinate set of entities (as in "John ate some bananas"). Specific nodes in the current fragment are also marked "new" or "possibly given". A "new" node may not be merged with any node in context, while a "possibly given" node may (and if possible will) be.

Generic, specific-and-new and specific-and-possibly-given nodes correspond approximately to some uses of KRL's specialization, individual and manifestation (when used to represent a "ghost") units respectively. However, a specific-and-possibly-given node, unlike a "ghost" manifestation unit, may not after all turn out to cospecify with an existing node.

(5) The context of identity[6] of a node is a list representing the "possible world", in a loose sense, in which the entity represented exists. An empty list means that the entity represented exists at "top level", i.e. in the world described by the story. The context of identity for the node `BREAK1-1` described in figure 4.3 is the desires of `HE1-1`. Contexts of identity are also set up for beliefs, statements, and possible futures, and may be nested, as in for example "He thought that Bill wanted to break a window".

The context of identity device could also be extended to make SPAR sensitive to quantifier scope, which at present it is not. The way this might be done is explained in chapter 10.

(6) The derivation of a node shows whether it is explicit (i.e. corresponds directly to a phrase in the text) or implicit, and the head word of the phrase in the text it originates from, directly or otherwise. The head word information is used only for diagnostic purposes.

---

[6]Not to be confused with the term "context" on its own, which always refers to the existing (non-current-fragment) part of the TMN. SPAR's "context of identity" does, however, correspond to some aspects of KL-ONE's "context" mechanism.

(7) The linguistic features of a node are mostly shallow, syntactic information. Some linguistic features are carried over from the analyser dependency structures for use as a short cut to paraphrase generation (see chapter 9), and some arise from relationships in the WSN. Most features are used only for generation; others are accessed during structure matching, as described in the next subsection. This rather ad-hoc aspect of SPAR's representation would benefit from being rationalised.

We have now examined all the types of information that can be attached to a TMN node. Since most or perhaps all of these types could be captured instead using Alshawi's highly flexible `Specialisation:/Corresponds:` assertions, the reader may wonder why this was not done, and even why the distinction between the WSN and TMN is necessary. The answer is that the strict separation of the WSN and TMN, and the use of different formalisms to represent them, is motivated by theoretical considerations - the desirability of a clearly defined text model in which model elements and story entities are ideally in one-to-one correspondence - and also by practical ones. Anaphor resolution is easier if the set of nodes and links modelling what is going on in the story is clearly delimited. For example, the feature list of a TMN node could be replaced by a set of `Corresponds:` assertions of the form

```
(Corresponds: CLAUSE to BREAK1-1 as CATEGORY to DUMMY)
(Corresponds: PRESENT to BREAK1-1 as TNS to DUMMY)
```

but then the TMN would contain nodes representing the concepts `CLAUSE` and `PRESENT`, which have no obvious counterparts in the story world, and certainly cannot participate in anaphor resolution and inference in the way that `BREAK1-1` can. Thus uniformity in the *form* of representation can only be achieved by ignoring important non-uniformities in *content*. SPAR's representation builds the most important semantic type distinctions (e.g. between word senses and story objects) into the syntax, thus much reducing the need to check at each stage that the type of processing being done is appropriate to the objects under consideration.

4.3.2 The structure matcher

SPAR's structure matcher plays a central role in the manipulation of the TMN which is necessary for anaphor resolution. It is oriented primarily towards assessing candidate antecedents for anaphors, but is also useful for making and matching common sense inferences, and, as we shall see in 4.3.3, for constructing current fragments. The description of the matcher given here is fairly abstract and mechanistic; the uses to which it is put will be described in more detail in the next subsection and in chapters 6 and 8.

Given two TMN nodes, the matcher compares them and decides whether they can be identified (or, sometimes, regarded as similar in some other way), and if so, what other pairs of nodes must be identified as a consequence. Because the representation is not canonical, the matcher must perform a limited amount of deductive linguistic inference in order to detect hidden similarities.

The structure matcher operates as follows. When asked to compare two nodes, it first invokes a node matcher, which determines whether the two nodes are compatible if links to other nodes are ignored. This involves inspecting most of the information associated with the nodes. The `Specialisation:` assertions are matched to check sense compatibility (and, when desired, satisfaction of the information constraint), but the Corresponds: links to other nodes are not accessed at this stage.

If the node match is successful, the structure matcher binds the nodes to one another so that they are provisionally regarded as equivalent. It may then call itself recursively on neighbouring pairs of nodes connected to the original pair by analogous Corresponds: links. For example, if the original pair of nodes is TELESCOPE1-1 and TELESCOPE1-2, and the relationships

```
(Corresponds: PARK1-1 to TELESCOPE1-1 as IN to DUMMY (E1))
(Corresponds: PARK1-2 to TELESCOPE1-2 as IN to DUMMY (E1))
```

hold, then the nodes PARK1-1 and PARK1-2 will be matched.

If possible, Corresponds: links specifying the same relationship are paired for submatching, as in our example where both specify the relationship IN. However, deductive linguistic inference will where necessary attempt to provide alternative pairings. This allows some flexibility in relation to:

(1) "vague" cases such as POSS (possessive) and WITH (attribute). For example, "John's book" may refer to a book that has virtually any relation to John: he may be holding it, reading it, or writing it.

(2) inclusion relationships: what is true of a set of entities may also be true of the individuals in it.

(3) lack of canonicality: for example the same action may be expressed using either GET or GIVE as a head primitive.

A successful submatch will return one or more node bindings which contribute to the overall result. However, the failure of a submatch may cause the whole structure match to fail. The effect of a failed submatch on the match as a whole depends on whether the matcher has been told to apply the information constraint: i.e. whether a successful match depends on all the information captured by one of nodes it is handed (typically representing an anaphor) also being captured by the other (typically representing a candidate antecedent). We saw in 2.2.3 that if this constraint is broken, the candidate antecedent can be much less plausible.

If the information constraint is in operation, then the overall match will only succeed if every Corresponds: assertion which is essential or intrinsic (in a sense we are about to define) to the description embodied by the anaphor node gives rise to a successful submatch. In a Corresponds: assertion of the form

```
(Corresponds: <CASE-FILLER> to <NODE> as <CASE-NAME> to DUMMY (<FLAG>)),
```

a <FLAG> value of E1 indicates that the assertion is "Essential" to the meaning of <NODE>. If <NODE> is the anaphor node in a structure match, the information constraint can only be satisfied if an analogous assertion involving the candidate node exists. If the <FLAG> is E2, the assertion is Essential to the meaning of <CASE-FILLER>. If the <FLAG> is E12, the assertion is Essential to both <CASE-FILLER> and <NODE>.

"Essential" flags on Corresponds: assertions often originate in the WSN, but can instead be derived from restrictive modifiers. The phrase "the man with the hat" would give rise to an assertion of the form

(Corresponds: HAT1-1 to MAN1-1 as WITH to DUMMY (E1))

which is intrinsic to the description of the man but not to that of the hat. This reflects the fact that in order to resolve the whole phrase as an anaphor without breaking the information constraint, we must identify a man who we know to have a hat; but "the hat" can be resolved without breaking the constraint merely by identifying a hat, whether associated with a man or not.

Relative clauses are usually treated as restrictive; when they are, they give rise to an E12 flag on the Corresponds: assertion connecting the element for the matrix NP and that for the relative clause.

If the information constraint is not in operation, only submatches resulting from Corresponds: assertions for case roles whose values are in some sense fixed and unique, such as the SUBJ and OBJE roles of action and event nodes, can cause overall failure. If two event descriptions have incompatible agents, they cannot be the same event. However, not all roles are unique in this way; for example, the phrases "the man with the hat" and "the man with the coat" can cospecify.

4.3.3 Construction of current fragments

Having examined the kind of information represented in the TMN, and the operation of the structure matcher, we are now in a position to see how TMN current fragments are actually constructed in preparation for anaphor resolution. The "fragment constructor" that carries out this task uses the structure matcher to compare WSN-derived information with information derived from Boguraev's sentence representations.

When SPAR processes a new sentence, it normally constructs one alternative current fragment of TMN for each analysis produced by the analyser.[7] Subsequently, as we will see in chapter 6, each fragment is independently compared with the context by resolving anaphors, and if there is more than one fragment, the one that fits into the context most naturally (in a way explained in later chapters) is accepted as correct.

Earlier (subsection 4.3.1) we looked in detail at the TMN node derived from the "break" in "He wanted to break a window". We will now see how the whole current fragment for this sentence is derived. The Memory assertions in this fragment are as shown in figure 4.4 (ignoring, for brevity, certain nodes and links derived implicitly from "window"). The parallels with the WSN assertions for BREAK1 shown in figure 4.1 should be apparent.

A fragment for a sentence reading is generated in the following three stages.

---

[7]Sometimes no fragments or several fragments result from a reading, because of limitations in the coverage of SPAR or the accuracy of Boguraev's analyser. Sometimes, because of the limitations of the analyser, obviously incorrect readings have to be altered or deleted by hand before SPAR is even invoked; however, such alterations are fairly rare, and in any case do not involve sidestepping theoretical problems of any relevance to anaphor resolution.

```
(Specialisation: HE1-1 of HE1)

(Specialisation: WANT2-1 of WANT2)
(Corresponds: HE1-1 to WANT2-1 as SUBJ to DUMMY (E1))
(Corresponds: BREAK1-1 to WANT2-1 as OBJE to DUMMY (E1))

(Specialisation: BREAK1-1 of BREAK1)
(Corresponds: HE1-1 to BREAK1-1 as SUBJ to DUMMY (E1))
(Corresponds: WINDOW1-1 to BREAK1-1 as OBJE to DUMMY (E1))
(Corresponds: BREAK1/GOAL-1 to BREAK1-1 as GOAL to DUMMY (E1))
(Corresponds: *INST-1 to BREAK1-1 as INST to DUMMY (E1))

(Specialisation: *INST-1 of *INST)

(Specialisation: BREAK1/GOAL-1 of BREAK1/GOAL)
(Corresponds: BREAK1/WHOLE-1 to BREAK1/GOAL-1 as OBJE to DUMMY (E1))
(Corresponds: WINDOW1-1 to BREAK1/GOAL-1 as SUBJ to DUMMY (E1))

(Specialisation: BREAK1/WHOLE-1 of BREAK1/WHOLE)

(Specialisation: WINDOW1-1 of WINDOW1)
```

Figure 4.4: Summary of current fragment for "He wanted to break a window"

(1) The "skeleton" of what will become the current fragment is constructed by creating a TMN node for each noun-arg and clause in the dependency structure, and a Corresponds: assertion linking two such TMN nodes for each case relationship in the dependency structure.[8]

(2) For each word sense occurring in a noun-arg or clause structure, a small network of TMN nodes is constructed using as a blueprint the relationships in which the word sense participates in the WSN, taking role inheritance into account.[9] The blueprint for a clause whose word sense was BREAK1 would be exactly the assertions in figure 4.1, since in this case there are no roles to inherit.

---

[8]Since dependency structures do not contain the same case names as formulas, but rather use the more detailed set developed by Boguraev, the names used in the Corresponds: assertions are derived by table lookup. No exact mapping from Boguraev's cases to Wilks' can be defined, but the approximation used by SPAR works well in practice.

[9]More formally, the relevant WSN nodes are those reachable from the original one by paths consisting of transitions from A to B in assertions of any of the forms

```
(Specialisation: A of B)
(Corresponds: A to B as X to Y)
(Corresponds: B to A as X to Y)
```

and terminating in a Corresponds: assertion.

(3) Pairs of nodes in the dependency-structure-derived skeleton and the small WSN-derived networks are handed to the structure matcher. In our example, the skeleton node for the BREAK1 clause would be matched with the node derived from BREAK1 in the WSN. The resulting bindings provide a measure of the "semantic density" (redundancy) of the reading; the more bindings of semantically compatible nodes that are made, the higher the density, because every such binding derives ultimately from a satisfied preference in a formula. Semantic density is one criterion used to decide between competing fragments, much as in PS and Boguraev's analyser.

Pairs of nodes bound by the matcher are then merged together, to make the current fragment fully connected. In our example, the explicit nodes BREAK1-1 and HE1-1 both originated as nodes in the skeleton fragment; WSN-derived nodes, bound to them by the matcher, were merged into them and then destroyed. The implicit nodes *INST-1 and BREAK1/GOAL-1 originated as WSN-derived nodes, and survived because they corresponded to nothing in the dependency structure and so were not merged into any skeleton nodes.

This merging can result in the semantic interpretations of pronouns being restricted in a way useful for subsequent anaphor resolution. In the case of the sentence "John drank it", a temporary node derived from the WSN element DRINK1/OBJE might be merged into the "it" node, which would inherit a Specialisation: link to LIQUID1.

Once merging is over, the status and context of identity of each node in the fragment are calculated. The algorithms used are ad-hoc and rather superficial, and would benefit from being rationalised; however, they give accurate enough results to support the other, more central aspects of processing.

If our sentence "He wanted to break a window" is the second sentence of a story beginning "John was angry", then once the current fragment has been built, anaphor resolution will (by methods described in chapter 6) identify our node HE-1 with the node JOHN1-1 in the context; no other nodes in the current fragment will be identified with nodes in context. When the current fragment is incorporated into the context, HE1-1 will be merged into JOHN1-1 and then destroyed. The merge will cause JOHN1-1 to inherit much of the information formerly associated with HE1-1, as indicated in figure 4.5.

4.4 Summary and evaluation

The aim of this chapter has been to show how SPAR's representation is designed to possess a number of characteristics desirable in a shallow processing anaphor resolver. Briefly, the required characteristics were that the formalism should:

(1) represent texts by a text model: distinct events and objects should be represented separately, with indications of their status and context of identity; and, conversely, each text model element should correspond to some story event or object, explicit or implicit;

(2) be fairly shallow, and deeper (more orthogonal and explicit) than Boguraev's input and output structures only to the degree required

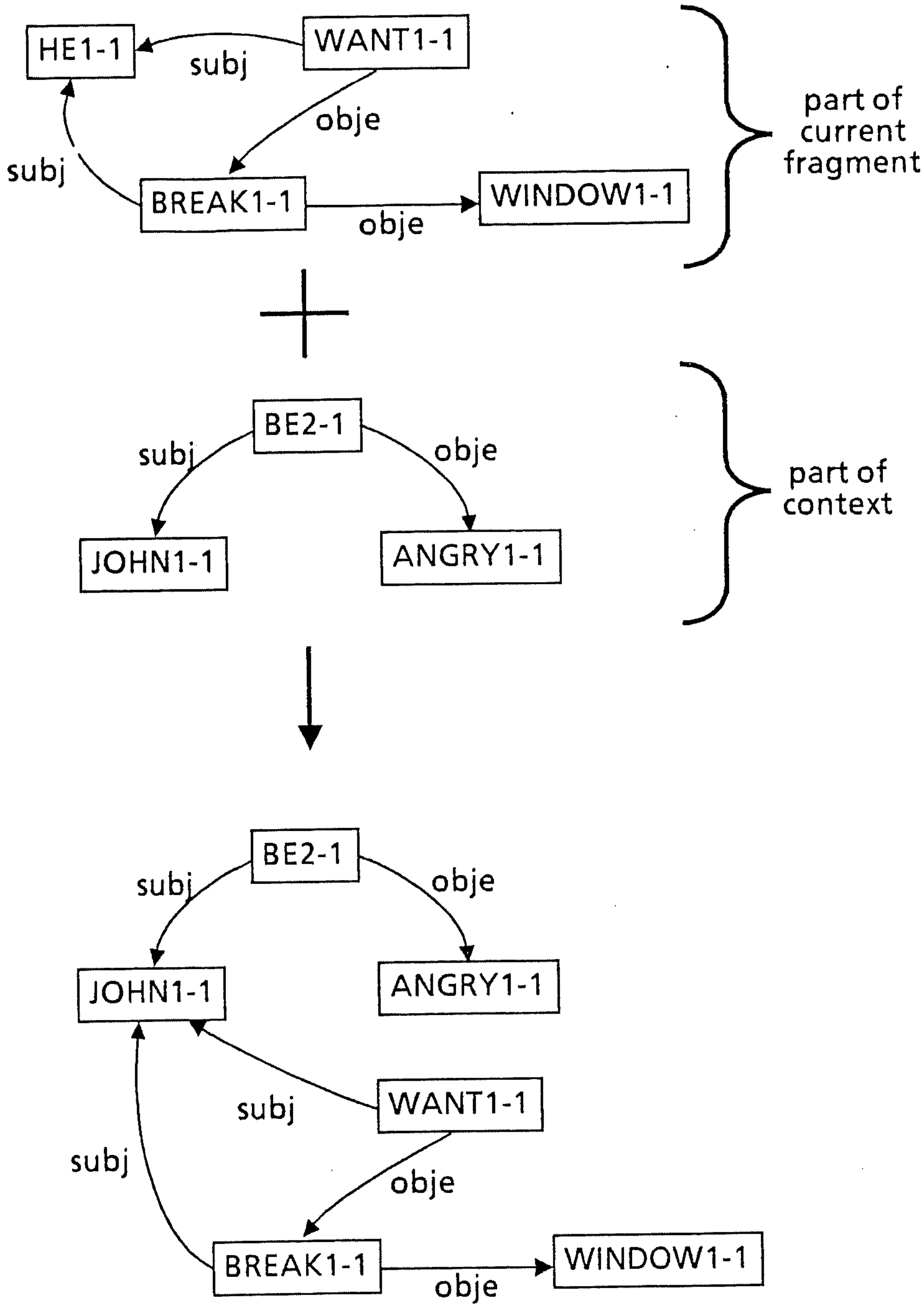


Figure 4.5: Before and after merging the current fragment into context

for limited CSI;

(3) be heterarchical and not hierarchical;

(4) be constructed by making maximum use of linguistic (specifically, word sense) knowledge.

(5) facilitate the structure matching required for anaphor resolution, allowing

(5a) reiteration and collocation to be treated uniformly; and

(5b) both compatibility and the information constraint to be checked (hence requiring a moderately well-defined semantics).

Requirement (1) is met by clearly separating linguistic knowledge (in the WSN) from knowledge of the current story (in the TMN), and storing statuses, contexts of identity and linguistic features within TMN nodes rather than uniformly by `Corresponds:` links. Requirements (2) and (3) are met by converting Boguraev's structures into heterarchical TMN structures as directly as possible: deriving one TMN element from each clause and (distinct) noun-arg, and leaving the case links between them essentially unchanged. Requirements (4) and (5a) are satisfied by augmenting the TMN with implicit nodes derived from word sense formulas. Requirement (5b) is met by using Alshawi's Memory formalism for the two networks, augmented by the "essential" flags on `Corresponds:` links.

The need to make maximum use of the information in word sense formulas brought to light evidence that Wilks' formula syntax, though designed to provide habitability, in practice makes it hard for formula writers to write what they mean. A new syntax, formally equivalent to the old, was proposed, in which all information was made explicit. It was argued that this new explicitness made formulas less natural-language-like but more habitable. In addition, the approximation to a formal semantics provided by the relatively trivial conversion to Memory formalism is a more reliable way of ensuring that a formula has the intended meaning than Wilks' [1977a] rules and descriptions of primitives.

The fact that SPAR's representational scheme meets the five requirements listed above means that it is well suited to the kinds of processing necessary for anaphor resolution by shallow processing methods; indeed, virtually every aspect of the representation was designed with anaphor resolution in mind. This makes it, like Alshawi's Memory on which it is based, considerably simpler and less powerful than other, more general representational formalisms. For this reason it is unlikely to be capable of supporting the inference required for many understanding tasks in the way that languages like KRL or KL-ONE may be able to. For example, the simple list format for contexts of identity, and (we will see in chapter 8) the way it is used in CSI, take little account of the difficulties of reasoning about beliefs, desires and "possible worlds". Similarly, it is perhaps rather naive to assume that every concept involved in a story can be treated as straightforwardly specific or generic. Perhaps the most awkward omission is the lack of any thorough treatment of multiple perspectives; for example, SPAR cannot represent cleanly, in the way that KRL can, the fact that the same event can be viewed as trip from one place to another from one perspective and as a visit to someone from another. However, we will see in later chapters that SPAR's simplified representation allows us to deal with a surprisingly wide range of anaphoric phenomena and types of inference.

5. Focussing and discourse structure

> "Like any powerful new idea, Focusing is not readily described in old terms. It moves us into unfamiliar territory...It is at once a manual and a philosophy."
>
> *Marilyn Ferguson, introduction to "Focusing" by Eugene T. Gendlin, Bantam Books, 1978*

In the last two chapters, we looked at the problem of defining a suitable meaning representation for an anaphor resolver. Chapter 3 was a backward look at the representations used by Wilks and Boguraev, while chapter 4 described that of SPAR. This chapter and the next form an analogous pair, discussing the phenomenon of focussing. The current chapter describes treatments of local and global focus and the related phenomenon of discourse structure, in order to provide the necessary background for chapter 6, which concerns SPAR's primarily focus-based anaphor interpretation process.

The structure of this chapter is as follows. First, in the light of the many and varied uses of the word "focus" in the literature, definitions of "focus" and related terms are given, and a distinction is made between global and local focus. Next, various anaphor-related treatments of global and local focus are reviewed. Finally, Sidner's theory of local focus, which is central to SPAR's operation, is described in some detail.

5.1 Focus, foci and focussing

At any point in a normal discourse, the speaker and hearer focus their attention more on some aspects of what is being discussed than on others. The speaker uses various devices to show that he is maintaining or switching his focus of attention, and the hearer attempts to interpret these and make the same switches. This focussing process is relevant to anaphor resolution because the hearer will, when possible, interpret anaphors as referring to focussed entities. The speaker assumes that the hearer will do this, and therefore uses anaphors for focussed entities in the expectation that they will be correctly interpreted.[1]

It is easy to see that the effects of focus of attention on anaphor resolution cannot be modelled purely in terms of recency of mention. In

(5-1) There was a zoo in John's town. <u>It</u> was small.

it is clear that it is the zoo, not the more recently mentioned town, that is being described as small, although either could equally well be small; and for

(5-2) The winning species would have a greater amount of competitive ability than the loser as far as that resource axis of the n-dimensional niche is concerned (e.g. <u>it</u> would be

[1]The problems that arise when the hearer is unable or unwilling to maintain the same focus as the speaker will not be discussed here.

more adapted to using that resource in that particular habitat).

Hirst [1981] claims that "there is no text which could replace the text after "it" and make a well-formed sentence in which 'it' refers to one of the more recent NPs" than "the winning species".

The task, then, is to define procedures that model the way focus of attention and anaphor resolution influence one another. But before considering various attempts to do this, some terminological problems must be cleared up. The term "focus" is used in several ways in the literature:

(1) The word or expression in a sentence which is the centre of phonological prominence.

(2) The state of the participants' attention at a given moment.

(3) The entity, in the world, on which the participants are centering their attention; or the element, in the hearer's or speaker's model of the world, representing that entity. ("Element" and "entity" are used very broadly here).

(4) The process of transferring attention from one set of entities to another as a discourse progresses. This is often called focussing.

Sense (1) will be called p-focus; it will be considered briefly later. Sense (2) will be called the state of focus or the focus state; it can be represented by associating focus values, drawn from some specified set, with elements and (we will see below) substructures in the discourse model. When it appears that one entity (or the model element representing it) is at the centre of attention and all others are of peripheral importance, we will call that entity or element the focus or the current focus in sense (3);[2] in section 5.4 we will refine this notion and introduce the terms the discourse focus and the actor focus. Sense (4), the process, will be called focussing. The term focus (without a determiner) will be used in a general sense which subsumes (2), (3) and (4).

Intuitively it seems that whether or not "the focus" exists in sense (3), some entities are more central at a given point in a text than others; in other words, that the focus state imposes some kind of ordering on entities. Some theories make this ordering explicit, while some do not. We will sometimes talk about one entity being more focussed or in focus than another; by this we mean that the more focussed entity is preferred as the referent of an anaphor when no syntactic, semantic or inferential factors bias the decision either way. Thus in (5-1), the zoo is more focussed than the town when "it" is read. Equating degree of focus with anaphoric preferences in this way serves to define more clearly what we mean by focus; the equivalence is not open to independent verification, since we are only interested in focus in so far as it affects anaphor resolution.

---

[2] In order to keep the terminology as simple as possible, our use of the term "the focus" is, strictly speaking, ambiguous between entities in the world and elements in the model. However the entity/element distinction is orthogonal to our concerns here; and since entities and elements are assumed to be in 1-1 correspondence, no confusion should result.

However, the focus value of an element is determined by more than the sequence of specifications of (i.e. references to) it in a discourse. A discourse tends to have a structure over and above the purely linear sequencing of its component sentences. A set of sentences may discuss one subject, and there may then be a switch to a different, though often related, subject, followed perhaps by a return to the original one. These switches are relevant to anaphor resolution because, as Grosz's work (see below) shows, they bring about wholesale changes in the focus values of the model elements involved. Grosz's work suggests that if the hearer is to recognise such switches and their consequences, he must have a discourse model with some degree of global structure, and also be able to focus on substructures within the model as well as on individual elements.

A discourse model does not exist in the hearer's mind in isolation from the rest of what he knows and perceives; it is related to the wider context of the hearer's knowledge, which has its own structure. This wider structuring also influences language interpretation, although it is perhaps less important for anaphor resolution than for other tasks such as word sense disambiguation.

We may therefore make a distinction, orthogonal to that between senses (2), (3) and (4) above, between <u>global focus</u> and <u>local focus</u>. Global focus is that aspect of focus which is determined through the broader structure of the discourse model and that of the knowledge to which it is related; it is therefore knowledge-based and only indirectly linguistic, and can involve substructures as well as elements. Local focus is the aspect of focus determined through the type and frequency of specifications of elements by phrases in the text; it is therefore purely linguistic, and applies only to model elements, not to substructures. The adjectives "global" and "local" are appropriate because the effects of local focussing tend to be shorter range, typically lasting for only a sentence or two.

All the treatments of focus discussed in this chapter can be seen as fleshing out and justifying part or all of this characterisation by specifying the nature of the discourse model, the set of possible focus values of an element or substructure, and how focus interacts with the interpretation process, particularly anaphor resolution.

5.2 Global focus and discourse structure

This section is concerned with global focussing. I shall start with Grosz's [1977,1978] work, the first and perhaps the most significant computational treatment of the subject, in order to provide a fuller characterisation of global focussing and its relation to global discourse structure. The global structures of the discourses Grosz was concerned with were determined by matching the discourses onto a predefined framework; next, therefore, we examine attempts, based on "coherence relations", to determine global structure (and hence maintain global focus) when no such framework is available. The section ends with a review of some treatments of focussing (in fact both global and local) as the cumulative effect of a number of largely independent factors.

5.2.1 The use of task structure: Grosz's theory

Grosz [1977,1978] formalized the notion of focussing for dialogues whose goal is the completion of a well-defined task. Using examples drawn from dialogues between an apprentice attempting to assemble an air compressor and an expert providing advice on how to do so, she argued that communication depends on the participants focussing their attention on the same small subset of their shared knowledge. Her theory was partly implemented in the TDUS system.

According to Grosz, the hearer can keep track of the (global) focus shifts intended by the speaker by following general linguistic clues, such as the use of certain conjunctions, and by using knowledge of the structure of the domain of discourse. However Grosz only considered the latter method in detail. The overall task of assembling the air compressor was analysed as a partially ordered hierarchy of subtasks involving particular events and objects. Grosz found that her dialogues had a structure similar to that of the task they related to. The predefined task structure therefore provided a framework for structuring individual dialogues without completely determining their form or content.

At any stage in a dialogue, the current global focus consisted of a number of "focus spaces", each corresponding to a particular subtask. The "active" focus space corresponded to the subtask currently under discussion, while an "open" focus space was associated with each of the superordinate (and therefore uncompleted) subtasks in the hierarchy.

When a FDNP (full definite noun phrase) was encountered, the search for its referent began in the active and open spaces, visiting neighbouring spaces in the hierarchy only when necessary. If a referent was found in a neighbouring space, that space would become active, and the spaces superordinate to it, possibly including some or all of the old active and open ones, would become or remain open. Thus focussing and anaphor resolution were mutually dependent (although a focus shift could also be provoked non-anaphorically, e.g. by a phrase such as "OK, what next?").

Grosz's work suggests that a fairly detailed representation of the structure of the domain of discourse is necessary for the maintenance of global focus, and therefore for the reliable resolution of non-pronominal noun phrases. However, other work discussed below suggests that significant progress can be made without such a representation, partly by exploiting the linguistic clues to discourse structure which Grosz largely ignored. Furthermore, it is possible that the dependence of anaphor resolution on a knowledge of discourse structure is proportional to the degree of structure, and is highest in very constrained discourses such as the ones Grosz considers; in Grosz's dialogues, the task constrains not only what things can be talked about but also the order of talking about them. In discourses with a less rigid structure, such as simple narratives or descriptions, the penalty for ignoring that structure may be less severe.

In any case, Grosz argues that whereas FDNP resolution is largely constrained by global focus, whose maintenance depends on detailed domain knowledge, pronoun resolution is more affected by local (or in Grosz' terms, "immediate") focus, which she sees as a linguistic phenomenon; the lack of detailed domain knowledge in a shallow processing system may therefore present fewer problems for resolving pronominal anaphors than for non-pronominal ones.

5.2.2 Coherence Relations

When no predefined framework for recognising discourse structure is available, one may attempt to recognise structure using only the linguistic form and content of individual utterances. One promising approach to this is based on coherence relations such as "elaboration" or "contrast". In this approach, it is assumed that every non-initial sentence in a text is related in one of a fixed number of ways to some earlier sentence. A number of researchers, including Halliday and Hasan [1976] in their analysis of "conjunction" as a type of cohesion, have attempted to provide a taxonomy of possible coherence relations. In text understanding based on coherence relations, the establishment of a relation between an existing sentence and a new one often presupposes, and in turn gives support to, particular choices of specifications for the anaphors in the new sentence.

Hobbs [1979] investigates what processing is required to establish that one of a given set of coherence relations holds between a given pair of sentences. He argues that much anaphor resolution simply "falls out" of the hearer's attempt to recognize coherence relations in what he hears by constructing inference chains. In Hobbs' framework, the second sentence of

(5-2) John can open Bill's safe. He knows the combination.

can be connected to the first by a coherence relation (in fact by a relation of Elaboration, which Hobbs defines) if and only if "he" is interpreted as John and "the combination" is interpreted as the safe's combination. The reader's assumption that the text is coherent leads him to accept those interpretations. (Note that "he" here cannot be resolved on grounds of plausibility alone; Bill is at least as likely as John to know the safe's combination).

The work of Lockman and Klappholz (henceforth L&K) [1980] is complementary to that of Hobbs. L&K assume the existence of an inference mechanism capable of verifying that a given coherence relation holds between a given pair of sentences, perhaps following Hobbs' methods. They propose a "contextual reference resolution algorithm" (CRRA) which would call such an inference mechanism at appropriate moments while processing connected texts.

L&K represent a text as a hierarchy of nodes, each of which is the semantic representation of a sentence. Nodes are linked by coherence relations. The hierarchy is initialised to a node for the first sentence. For subsequent sentences, the CRRA works by initially hypothesising a coherence relation between the new sentence and the most recently added one. If a plausible relation (with concomitant anaphor resolutions) is discovered by the inference mechanism, it is accepted, and the new node is connected to the hierarchy by that relation; if not, the CRRA moves through the hierarchy in a defined way, considering other nodes and accepting the first plausible relation found.

L&K's theory has some interesting similarities to Grosz's. Both represent texts hierarchically; nodes considered early by the CRRA are analogous to active or open focus spaces; and focussing (i.e. the CRRA operation) and anaphor resolution are mutually dependent, as in Grosz's theory.

However, the CRRA appears to be potentially more flexible than Grosz's approach, because it constructs the text hierarchy dynamically, while Grosz's algorithm can only deal with discourses conforming to a predefined task structure. But a reliable inference mechanism of the type that L&K assume for assessing hypothesised relations is probably beyond the state of the art; and any unreliability, for example rejecting a correct hypothesis, is liable to result in a great deal of unnecessary computation and perhaps eventually a wildly incorrect attachment.

5.2.3 Cumulative treatments of focus

Both Grosz's and L&K's theories treat focus solely in structural terms: the focus state of an entity depends on its position in a representation of the discourse structure. In contrast, a number of researchers have viewed focussing as the cumulative effect of several largely independent factors (perhaps including some notion of discourse structure); the focus state of an element is then essentially the sum of several numbers. An early approach along these lines was that of Kantor [1977], who developed the idea of "concept activatedness", where "the more activated a concept is, the easier it is to understand an anaphoric reference to it". Kantor characterised activatedness as the net effect of factors such as discourse topic, the syntactic positions of earlier references, and syntactic parallelism. However, since his analysis is a largely qualitative one, its predictions are difficult to assess.

Kantor's general approach is echoed in the "principle of factor cumulation" used in the Con^{3}Tra project (Pause [1984]). This principle states that "various (more or less certain and significant) clues from a given text work together in supporting a certain hypothesis about a possible antecedent for an anaphor and that finally the correct hypothesis - indicating the intended antecedent - outweighs the other ones". An assumption very like this in fact underlies the shallow processing hypothesis: if in a considerately-written text various factors work together, rather than against each other, to indicate an antecedent, then a less than ideal treatment of the particular factor of common sense inference should not often lead to the wrong antecedent being accepted.

A third cumulative approach is that of Alshawi [1983], whose Capture system resolved anaphors and other ambiguities in texts as a means to generating database creation statements. The "context activation" (roughly, degree of focus) of each candidate referent in the text representation at a given moment was defined as the sum of a number of "context factors", such as sentence recency and syntactic emphasis, which decayed over time. For a singular definite anaphor, the most activated syntactically and semantically plausible candidate would be accepted; resolving plural anaphors could involve threshold application as well.

Alshawi's context factors represented a wide range of context influences in a simple and uniform way. It seems likely that because of its flexibility, the context factor idea could be extended to other domains, tasks and text types. Alshawi is careful to stress that the importance of his work lies in its possible practical applicability rather than in any specific theoretical claims about language, since the behaviours of the various context factors were arrived at on a trial-and-error basis rather than as consequences of a comprehensive linguistic theory. However, the test set of over thirty texts which Capture processed correctly is impressive. While it does not conclusively demonstrate the correctness and full generality of the mechanisms involved, it does suggest that global as well as local focus can, for some tasks and domains at least, be maintained without relying heavily on the detailed domain knowledge used by Grosz or on rule-based inference mechanisms.

5.3 Local focus and sentence structure

At the beginning of this chapter, local focus was defined as the aspect of focus determined by how often and in what way various elements are specified by phrases in a text. This section goes into more detail, covering first some treatments of local focus and topic in theoretical linguistics, and then some approaches to anaphor resolution which do not represent local focus explicitly but define it implicitly by using sets of rules which suggest antecedents for pronouns in a given order. These rules embody some of the insights discussed in 5.3.1. (Sidner's work, which will be discussed in section 5.4, can be seen as a related approach in which focus is made explicit.)

5.3.1 Focus and topic in sentences

Much work on sentence organisation has been based on the distinction between "given" information, which is derivable from the previous context, and "new" information, which is not. For Lyons [1968], for example, the topic of the sentence is what the sentence is "about"[3] and is usually given, while the "focus" is often defined as in some sense the centre of phonological prominence, and is therefore usually new (e.g. Halliday [1967], Chomksy [1971]). Neither concept is to be identified with our notion of focus, because the items that a sentence puts most into focus (in the sense of being preferred as referents) can be either given or new. I will therefore use the term p-focus (prosodic focus) to refer to the linguists' concept. However, it is usually the case that the topic is more focussed than any other "given" information and the p-focus more focussed than any other "new" information, and therefore treatments of both concepts are of interest.

Focussing is affected (or effected) by a range of prosodic, syntactic and semantic choices; but because we are concerned only with written language we will discuss prosody only where it is essential to the views reported.

(a) Focus and syntax. A number of marked syntactic constructions determine p-focus and direct attention, and therefore focus, away from the topic and towards the p-focus, which is underlined in the following examples (however, whether the p-focus is also "the focus" is a function of context).

(5-3) There was a zoo in John's town. ("There-insertion")

---

[3]This definition is not as vague as it may seem; Reinhart [1982] introduces an "aboutness test" for identifying topics. Note also that Halliday [1967] uses the term "theme" for roughly the same concept as Lyons' "topic"; however, we will use "theme" in a different sense.

(5-4) It was his best suit that John wore to the dance last night. ("Clefting")

(5-5) What John wore to the dance last night was his best suit. ("Pseudo-clefting")

Left-extraposition (typically of the topic) has a similar effect to pseudo-clefting, making some of the non-extraposed items more focussed.

(5-6) To the dance last night John wore his best suit.

In unmarked cases, nuclear items (subjects and objects) tend to be more focussed than items such as prepositional objects. The subject of a passive sentence is usually especially strongly focussed.

(b) Focus and semantics. Gruber [1976] and Anderson [1977] use a semantic notion of theme to account for a number of otherwise unexplained regularities in English, such as that between the transitive and intransitive forms of "break". Neither author provides a specific and comprehensive definition of theme, arguing instead for its identification of a variety of different grounds; for example, if a sentence describes a change to an entity, that entity is the theme. The theme of a sentence often appears as the direct object, or, for an intransitive verb, as the subject.

The importance of the concept of theme for our purposes is that themes are often more focussed than other participants: that is, they are preferred as referents of subsequent anaphors. Although the concept is not defined for all verbs, it is sufficiently useful for Sidner's PI rules to apply it, and SPAR is normally able to recognise the theme of a clause by examining the head primitive (see 4.3.1) of the definition of its verb.

5.3.2 Rule-based approaches to local focus

A number of researchers have formulated sets of rules for suggesting candidate antecedents for pronouns in a defined order. This order can be seen as defining a focus ordering on the candidates' specifications; more focussed candidates are suggested first. Because the rules consider only the linguistic form and role of antecedents, the type of focus modelled is local and not global.

Winograd's SHRDLU system [1972], could resolve a wide variety of anaphors in its dialogues about its blocks world, including definite and indefinite pronouns and verbal substitutes ("do it/that"). It was impressive in its avoidance of simple recency criteria, in the range of anaphors covered, and in the way anaphor resolution was coordinated with other aspects of processing. However, as has often been remarked, its performance was partly due to the finite nature of the micro-world within which it operated; and Winograd recognised that the plausibility heuristics used were "fairly arbitrary". The focussing phenomena they were meant to cover are treated more rigorously and coherently by Sidner.

Hobbs [1976] developed a purely syntactic algorithm for definite pronoun resolution which selected antecedent noun phrases from surface syntactic parse trees for sentences in a text. The algorithm performed surprisingly well, achieving 88% accuracy on 300 pronouns occurring in three (real) texts. This figure was increased to 92% when selectional restrictions were allowed. The algorithm is too long to reproduce here, but it essentially preferred candidate antecedents in the current sentence to those in earlier ones, less deeply embedded candidates to more deeply, and for those in the current sentence, there was a tendency to favour candidates near the pronoun in the parse tree over those further away, and to favour anaphora (in the strict sense) over cataphora.

It might seem that because of its high accuracy, Hobbs's algorithm is the last word in anaphor resolution by shallow processing. In particular, one might argue that no shallow-processing system whose accuracy is less than 88% has any claim to significance. However Hobbs argues that "there is every reason to pursue a semantically based approach", for a number of reasons. Firstly, no purely syntactic approach can hope to do (much) better than his own, which "offers no hope of a total solution"; his algorithm, while impressive, is a dead end. In particular, it offers no help in recognising which 12% of its decisions are the wrong ones. Secondly, although his algorithm works, it is not explanatory; he argues that his semantic approach (to be discussed in chapter 7) is more appealing since it "depends on very fundamental properties of language". Thirdly, semantic processing must be done anyway in the analysis of texts, and a purely syntactic anaphor resolver is therefore anomalous.

The real importance of Hobbs's algorithm is that it combines reasonable accuracy with computational cheapness; it therefore seems sensible to use it as one component in an anaphor resolver. SPAR in fact uses it to impose a weak "focus" ordering on intrasentential candidate antecedents, since Sidner covers thoroughly only intersentential anaphora.

Guenthner and Lehmann [1983] present a set of six preference rules, expressed, like Hobbs', in terms of surface syntactic parse trees for the current sentence and earlier ones. These rules are used to resolve pronouns in the restricted context of relational database query dialogues. The authors stress that the rules are only intended to apply to their particular type of dialogue, but express the hope that they will be extendible to other types.

Closer examination reveals striking similarities between the preference rules of Winograd, Hobbs (as implied by the algorithm) and Guenthner and Lehmann. All express preferences for candidates in more recent sentences and favour subjects over objects and both over candidates outside the nucleus. We will see below that Sidner's rules, which are more complex, more rigorously justified, and in some ways more comprehensive, also reflect most of these preferences.

5.4 Sidner's theory of local focussing

In chapter 1, I summarised Sidner's theory of local focussing. This section examines some aspects of Sidner's work in more detail in order to prepare for the description of SPAR's anaphor-resolving capabilities in later chapters.

Sidner [1979,81,83] assumes that, at a given point, a well-formed discourse is "about" some entity mentioned in it. This entity is the discourse focus.[4] As the discourse progresses, the speaker may maintain the same discourse focus or may focus on another entity. A change in discourse focus, or the lack of a change, are signalled by the linguistic choices the speaker makes, particularly his use of various anaphoric expressions. The hearer must use his knowledge of certain rules governing these choices and of the subject-matter of the discourse to interpret anaphors and follow changes in discourse focus.

In describing her work, I will follow Sidner in using the term "the focus" to mean "the discourse focus" rather than the "actor focus" which will be introduced in due course.

After setting out Sidner's computational framework, I will describe her rules for resolving FDNPs and third-person definite personal pronouns. I will not discuss her treatment of "co-present foci", as exhibited by "the one...the other" and "this...that", because it introduces no radically different ideas and because anaphors of this type are uncommon in simple stories. I will then discuss Grosz, Joshi and Weinstein's [1983] theory of "centering", which is in some ways an alternative to Sidner's, and finally evaluate Sidner's theory and compare it with Grosz et al's work and with other work described in this chapter.

5.4.1 Sidner's processing framework

Sidner's apparatus is as follows. The state of focus at a given point in the text is represented by the contents of six focus registers. The discourse focus (DF) and actor focus (AF) registers each contain the representation of a single entity mentioned in the text; the potential discourse focus (PDF), potential actor focus (PAF), discourse focus stack (DFS) and actor focus stack (AFS) registers each contain a list of zero or more entities. The uses of these registers and the rationale behind them are explained below. Sidner formalises the anaphor interpretation process in various algorithms to be applied in turn to semantic representations of input sentences:

(1) The expected focus algorithm is applied only to the first sentence in a text. It uses some of the syntactic and semantic criteria described in 5.3.1 above to select an expected discourse focus which may or may not be confirmed in subsequent sentences. The other entities mentioned in the sentence (whether by noun phrases or clauses) are alternative or potential foci. The DF register is set to

---

[4] However Sidner makes it clear [1983,p279] that the discourse focus is defined not as what the discourse is about but as what element her algorithms select as the discourse focus. The latter is of course an attempt to capture the former.

The many footnotes in this section are intended mainly to point out the inevitable simplifications in my overview of Sidner's very detailed theory. They are not essential to gaining a general understanding of Sidner's work.

the expected focus and the PDF register to the potential foci.[5]

(2) For non-initial sentences, an anaphor interpretation algorithm is applied to each anaphor. There are different algorithms (Sidner [1979] lists seven) for anaphors of various grammatical types and in various roles in the sentence representation. Each algorithm is a discrimination net containing ten or so rules; those for definite pronouns are called PI (pronoun interpretation) rules. Each rule in the algorithm appropriate to an anaphor suggests one (or sometimes several - see later) specifications (i.e. elements in the text representation) for it according to what the focus registers contain. The suggested specification is assessed by an inference mechanism (which Sidner assumed to exist) which looks for any resulting contradictions. The first suggestion not giving rise to a contradiction is accepted.

(3) After anaphor interpretation, a focus update algorithm is applied which updates the focus registers, taking the results of anaphor interpretation into account. If the DF changes, the old DF is pushed onto the DFS, or, if the new DF is already in the DFS, the DFS is popped. Whether the DF changes or not, the PDF list consists of representations of every entity mentioned in the current sentence other than the DF itself. The AF, PAF and AFS registers, which we will discuss later, are updated analogously, except that only animate entities can be held in these registers.

The next sentence is then processed by stage (2).

Thus in Sidner's theory, definite anaphors are seen as signals which tell the hearer what elements are in focus and in what registers, and the focus state, as defined by the six focus registers, in turn partly determines the interpretation of definite anaphors. Sidner supports her theory by presenting evidence for five claims which can be summarised as follows.

(C1) Focussing provides a means for determining whether a definite noun phrase specifies something already mentioned in the discourse, specifies an element associated with the discourse, or specifies something outside the discourse.

(C2) Focussing (together with sentence syntactic and semantic information) distinguishes pragmatic (discourse) anaphors from bound-variable and intrasentential ones.

---

[5] Whether the DF is "confirmed" or only "expected" is of no procedural importance except when a sentence with no anaphors is encountered. When this happens, the DF will remain unchanged if it is confirmed; otherwise the DF register is emptied and a list of "focus sets" is created as a temporary alternative to the normal focus registers. However, the texts which Sidner [1979,pp76-77] uses to argue for this special mechanism can be dealt with equally well without it by a small modification to her algorithms. Focus sets, and the confirmed/expected distinction, will therefore be ignored from now on.

(C3) The idea of focussing is the basis for a set of algorithms which (together with a suitable inference mechanism to assess suggestions) determine the specifications of pragmatic anaphors.

(C4) Focussing reduces the amount of inference to be done, since inference need only attempt to confirm or reject a suggested specification, rather than suggest one itself.

(C5) The data structure representing the current focus indicates what items may be associated with the focus. The range of phrases that may be used to mention those items determines certain necessary characteristics of the knowledge representation.

The evidence Sidner presents for these claims will be discussed in the rest of this section.

5.4.2 Interpreting full definite noun phrases

Sidner largely substantiates claims (C1), (C5) and, for non-pronouns, (C3), by a set of rules which apply to the focus registers. These rules assume a hierarchical/associative knowledge representation which provides for generic nodes, representing classes of objects or events such as "meeting", and specific nodes, representing instances of those classes (e.g. the meeting I am going to tomorrow).[6] Nodes may be linked by is-a (instance) and role-filler links (such as the "time" role for a meeting). Sidner cites Fahlman's [1979] NETL as an example of a suitable representation; SPAR's representation also meets Sidner's requirements.

Sidner lists several ways in which a FDNP may derive its specification from a focus (current, potential or stacked).

(1) Cospecification (1): the FDNP and focus cospecify if the FDNP has the same head as the focus and introduces no new information. Example: "a small office...the office".

(2) Cospecification (2): the FDNP and focus cospecify if the FDNP has a head which lexically generalizes that of the focus and no restrictive postmodifiers. Example: "a ferret...the animal".

(3) Associated Specification: if the FDNP names an element associated with the focus in the hierarchy, either directly or by role inheritance, then the FDNP specifies that element. Any inferences made in the course of establishing the association must be analytic (logically true). Example: "a meeting...the participants".

(4) Inferred Specification: as for associated specification, but establishing the association involves non-analytic inferences.

---

[6]Sidner also states that "prototype" nodes are required, one per generic class, to represent a typical member of the class. However, prototype nodes are not needed for SPAR's processing, and will therefore not be discussed further.

Example: "the dead heiress...the murderer".

(5) Set-element Specification: if the focus is a set, and the FDNP is singular and has the same head as the focus and some additional modifier(s), then the FDNP specifies an element of the focus. Example: "a herd of elephants...the elephant with the limp".

(6) Computed Specification: if the FDNP has an ordinal modifier, the same head as the focus, and no relative clause modifiers, the specification of the FDNP may be computed from that of the focus. Example: "a meeting...the last meeting but two".

The algorithm for determining an FDNP's function specifies an order for testing for particular relationships between the FDNP and particular (actual, potential or stacked) foci. Not all combinations are allowed. If no test yields a positive result, the FDNP is assumed not to be anaphoric. Sidner applies the algorithm to a wide range of examples and shows that it makes correct predictions. Those parts of it implemented in SPAR show similarly reliable performance when the required knowledge is present.

5.4.3 Interpreting definite pronouns

For definite pronouns, as for FDNPs, Sidner develops algorithms to suggest specifications. These specifications, like those for FDNPs, are to be assessed by an inference mechanism; however, the application of selectional restrictions and configurational constraints such as c-command plays a more prominent part in this inference for pronouns than for FDNPs.

Sidner begins by arguing for a recency rule and a basic rule. The basic rule is similar to that for FDNPs, and states that the focus should be suggested first, followed by each potential focus in turn if necessary. This rule is modified as the argument develops. The recency rule applies, before the basic rule, to sentence-initial subject pronouns only,[7] and states that if there is a potential focus which was specified by the last constituent of the previous sentence, that potential focus should be suggested first. Sidner admits that the recency rule "makes focussing seem somewhat ad hoc" and sees no clear reason for the phenomenon, but claims to have observed it to be consistently accurate. However its inclusion in SPAR led to considerable inaccuracy, and indeed

---

[7]Sidner in fact states conflicting conditions of application on different occasions; e.g. subject position [1979,p144]; sentence-initial [1979,p147]; subject position and sentence-initial and second sentence of text [1981,p230].

Sidner's examples [1979,p145] hardly lend support to it.[8] The recency rule is therefore not implemented in SPAR.

If pronouns were resolved using the basic rule alone, they would be treated just like FDNPs (except of course that the only specification relation they can participate in is cospecification). However, Sidner argues that a separate actor focus is needed to account for the behaviour of pronouns. The actor focus is defined as the agent of the most recent sentence that has an agent. Other animate specifications in the most recent sentence are potential actor foci; if the actor focus changes, the old value is pushed onto the actor focus stack.

Within the framework developed so far, Sidner suggests separate algorithms for pronouns which occur as agents, as possessives and as neither (henceforth "normal"). They are too long to reproduce in full here. However they may be summarised as follows (ignoring their use of the recency rule).[9]

> (1) For singular normal pronouns, the suggestions are: first discourse focus; then conversationally associated element of discourse focus (i.e. an already-mentioned element associated with the discourse focus); then potential discourse foci in textual order; and lastly actor and potential actor foci.
>
> (2) For singular agent and possessive pronouns, the relative preferences for discourse focus, actor focus and potential actor foci are much less clear. If several are acceptable, the usage may be judged ambiguous, or there may be a slight preference in some direction.

---

[8] For example, for

(5-7) Mary is giving a surprise party at Hilda's house. It's at 340 Cherry St

Sidner claims that the "it" cospecifies with "Hilda's house", and so the recency rule must intervene to prevent the expected focus of the surprise party being suggested. However my intuition is that it is the party which is at 340 Cherry St; Hilda's house is 340 Cherry St. In any case this seems to be an example of a phenomenon which Sampson [1983] claims is common: the correct antecedent is unclear, but the choice makes little difference to the underlying content of the text.

Secondly, for

(5-8) Fill the pan with the cake mixture. It will be slightly lumpy

Sidner implies that the recency rule must override the basic rule to prevent the pan being suggested as specification of "it". However this seems unnecessary; although selectional restrictions may not rule the pan out (pans can perhaps be lumpy) the stipulation of a lumpy pan in a recipe is so odd that an inference mechanism should reject it.

[9] The summary here is of the rules in Sidner [1979]. Sidner [1981] contains a description of some slightly different rules. However SPAR uses the 1979 rules because of their greater explicitness and completeness.

(3) Plural pronouns are dealt with like singulars, except that suggestions for combinations of focus elements (e.g. actor focus plus potential actor foci) occur at various points. One advantage of Sidner's framework is that it provides a partial solution to the difficult problem of computing specifications of plural pronouns where no single antecedent phrase exists.

(4) In most cases candidates are suggested one at a time to be assessed by "normal mode" inference, thus imposing strong preferences between them. Sometimes, however, several are suggested at once and "special mode" inference must choose between them; the focus preferences (if any) between such candidates are weak.

The use of these algorithms, without the jettisoned recency rule, may be illustrated by the following text (the example is mine).

[1] Susan was given a computer.

Expected focus calculation: expected DF is the computer, as theme. PDFs are Susan and the giving event, in that order. AF is null, since no agent is mentioned; PAF is Susan.

[2] She used it to write business software.

PI rule application: (1) She: agent rules predict Susan as PAF (there being no AF). Approved by inference. (2) It: normal rules predict the computer as (expected) DF. Approved by inference.

Focus update: Computer confirmed as DF. PDFs are the business software, Susan, the writing and the using in that order. AF is Susan.

[3] Thousands of people bought it.

PI rule application: It: normal rules suggest computer as DF. Rejected by inference, since thousands of people are unlikely to buy one computer. Normal rules then suggest the business software as first PDF. Approved by inference.

Focus update: Business software becomes DF. Computer pushed onto DF stack (previously empty). PDFs are the thousands of people and the buying. AF becomes the thousands of people. No PAFs. Susan pushed onto AF stack (previously empty).

[4] She became extremely rich.

Pronoun interpretation: She: Agent rules (assuming that "becoming" is a deliberate action) suggest thousands of people as AF; rejected on agreement grounds. Agent rules then suggest Susan as top of AF stack; this is approved by inference.

Focus update: DF becomes Susan. Business software pushed onto DF stack. PDF is "becoming" event. AF is Susan; no PAFs. AF stack is empty since Susan was popped from it.

The behaviour of the PI rules supports Sidner's claim (C4) that focussing reduces the complexity of inference. In the above text, the inference mechanism was only required to approve or reject suggested specifications for pronouns, rather than to infer those specifications itself, a far more complex and expensive task.

5.4.4 An alternative theory

Grosz, Joshi and Weinstein (henceforth GJW) [1983] give an account of anaphora which they contrast with Sidner's. Their notions of (backward) centre and forward centre correspond roughly to Sidner's discourse focus and potential discourse focus respectively. They propose the following rule as a constraint on the speaker:

(5-9) If the centre of the current utterance is the same as the centre of the previous utterance, a pronoun should be used.

This rule is not hard and fast; however, violations of it force the hearer to draw additional inferences to explain the violation. If no explanation can be found, the discourse is perceived as ill-formed. For example sentence [2] of the following sequence provokes a violation if John is the centre at the outset:

(5-10) [1] He called up Mike yesterday. (he=John)
[2] He was annoyed by John's call.

GJW note that Sidner dealt with multiple pronouns in a single sentence by the introduction of an actor focus, and imply that their simpler framework can handle multiple pronouns without such a device. They take Sidner's [1981] text

(5-11) [1] I haven't seen Jeff for several days.
[2] Carl thinks he's studying for his exams.
[3] But I think he went to the Cape with Linda.

where the pronoun "he" in [3] is potentially ambiguous between Carl and Jeff, and observe that "on our account, Jeff is centre after [2] and there is no problem", whereas in Sidner's framework, "fairly special rules" are necessary to make the choice correctly.

However, there are problems with this argument. The choice of example is unfortunate; Sidner [1981,p223] uses (5-11) not to argue for an actor focus (which she does using other examples) but to discuss some of the problems an actor focus mechanism raises. In fact her basic rule accounts for all the pronouns in (5-11) without any extra "special rules"; it is therefore unsurprising that the centering mechanism can also deal with it. In any case, GJW set out to state constraints on the speaker's choice of words, not the hearer's choice of interpretation. This means that their claim that rule (5-9) allows, without the need for a separate actor focus, usages such as the replacement of sentence [3] in (5-11) by

[3'] He thinks he studies too much.

while true, is misleading. Although it ***allows*** this usage, it does not, at least on its own, ***predict*** the specification of either pronoun. Thus although the GJW [1983] framework is simpler than Sidner's, GJW [1983] presents no evidence that it has anything like the same predictive power for interpretation.[10]

5.4.5 An evaluation of Sidner's theory

Any theory which aims to restrict and order the set of candidate specifications considered for anaphors may be evaluated on a number of different grounds. These include:

(1) Coverage: to what range of anaphors and related phenomena does the theory apply?

(2) Accuracy: for what proportion of anaphors does the theory make predictions which lead to an appropriate specification being accepted?

(3) Simplicity: how simple are the structures and mechanisms involved?

(4) Efficiency: Inference is expensive and difficult. How far does the theory eliminate or reduce the need for it?

Sidner's theory is satisfactory in most of these respects.

(1) Coverage. Sidner analyses FDNPs and a wide range of definite pronouns. However, phenomena not covered include:

(a) Parallelism. An anaphor resolver sometimes needs to take into account syntactic and semantic parallels between the sentence containing the anaphor and an earlier one. Sidner recognises this problem, but does not attempt to solve it because "proper computational recognition of parallelism is still beyond the state of the art" [1981,p229]. None of the other theories we have examined tackles parallelism seriously either.

(b) The influence of global focus. Sidner does not attempt to cover anaphoric behaviour when there is a change of global focus.

(c) Interactions between multiple anaphors in the same sentence. If the anaphor interpretation rules are applied independently to each anaphor, the resulting predictions may be individually acceptable but collectively unacceptable, because of e.g. a c-command violation. Sidner recommends the principle "choose a pronoun to co-specify with the focus before the choice of defnp [=FDNP]". However she

[10] In this context the work of Kameyama [1985] is interesting. Kameyama uses the centering framework to formulate constraints on both the production and interpretation of anaphora in Japanese. Although she avoids introducing an actor focus, she shows that an adequate account of anaphor interpretation involves far more than a simple preference ordering on backward and forward centres. It seems likely that an attempt to formulate a centre-based algorithm for interpreting English anaphora would involve a corresponding increase in complexity.

does not discuss clashes between pronouns. This problem, and SPAR's solution to it, are discussed in detail in the next chapter.

(2) Accuracy. Experience with SPAR has shown that once the recency rule is abandoned, Sidner's algorithms yield highly accurate results in the absence of phenomena (a)-(c) above. The only exception is that the algorithms always prefer a focus-derived specification (originating from an earlier sentence) to one originating in the current sentence;[11] this can lead to wrong results, as in

(5-12) John walked into the room. He told Bill that someone wanted to see him.

where the intrasentential candidate Bill is at least as plausible as the focus John. Interestingly, the algorithms that were discussed in 5.3.2 all prefer intrasentential candidates over others.

Sidner tested the accuracy of her theory in two partial implementations: in PAL (Personal Assistant Language Understanding Program), and in TDUS, along with Grosz' theory of global focussing (see 5.2.1 above). Details of these implementations are given in Sidner [1979]. In PAL, a simplified version of the PI rules, without an actor focus mechanism, was used. In TDUS, Sidner's rules were used to resolve only pronouns and not FDNPs. The rules reportedly performed well in both systems: in PAL, because the the dialogue was typically only one goal deep and so no shifts of global focus occurred, and in TDUS, because global focus shifts had already been detected by the time the rules were applied.

(3) Simplicity. Sidner's rules are not simple; their expression in abbreviated form covers twelve pages in Sidner [1979]. The complications may partly arise from her placing of the agent of a sentence after oblique (prepositional) objects in the expected focus ordering, and a corresponding de-preference for agents in discourse focus updating. Kameyama [1985] shows that the examples Sidner uses to argue for her expected focus ordering provide equally good support for an ordering in which the subject (which is often the agent) is preferred to oblique objects. It would appear that if the theme and syntactic subject are given approximately equal preference, and are preferred to oblique objects, the need for an actor focus and a separate agent-pronoun algorithm is much reduced if not eliminated.

(4) Efficiency. A theory which presents an inference mechanism with candidate specifications or antecedents one at a time will enable simpler and more efficient processing than one which presents several at a time. In the former case, inference need only check for a contradiction; in the latter case,

---

[11]Sidner [1979,p148] discusses the sentence "Shem loves his sister" occurring in mid-discourse, and states that "the focus rules force the freely cospecifying pronoun to cospecify with a focus if one is established; the focus is therefore the source of cospecification rather than some other noun phrase within a sentence". However in Sidner [1979b,p249] she says "It appears that the focus and potential focus ought to be checked for coreference to [he/she] pronouns before sentential coreference rules are used. However, ***further experimentation with such cases is needed*** to confirm this aspect of coreference." (my emphasis). It may therefore be that intrasentential candidates are ignored by her theory rather than wrongly treated.

it must find a reason to prefer one of the candidates. The fact that Sidner's algorithms often present candidates one at a time is a major advantage of her theory. However, as we have seen, Sidner recognises that the mechanism must "be capable of a special judgment when given one actor and one potential actor; it must weigh its findings, and choose one of the two candidates as superior". Although this judgment is "special" it is quite frequently needed, especially, as we will see, when intrasentential candidate antecedents are properly treated. Even so, Sidner's algorithms do seem to go as far in the direction of presenting candidates one at a time as is consistent with accurate predictions.

Thus in summary, Sidner's theory has very good coverage and efficiency. Its accuracy is in practice also good, except for intrasentential anaphora. Its simplicity leaves something to be desired. However, as a tool for practical anaphor resolution it is the most useful and complete theory of local focussing yet formulated.

6. Coordinated anaphor resolution in SPAR

> "The most heterogeneous ideas are yoked by violence together."
>
> *Samuel Johnson, "Lives of the English poets"*

This chapter describes SPAR as a whole, showing how components exploiting syntactic, semantic, local-focus and world knowledge fit together and presenting solutions to four problems of coordination which arise when the attempt is made to apply Sidner's framework to story processing. These problems are:

(1) How should anaphor resolution be coordinated with the resolution of word sense and structural ambiguity?

(2) How should the different knowledge sources - local focus, syntax, semantics and world knowledge - be coordinated so as to achieve accurate results as efficiently and simply as possible?

(3) How should the consideration of candidate antecedents from earlier sentences be coordinated with the consideration of intrasentential candidates (ISCs)? More specifically, how can Sidner's PI rules be extended to predict ISCs at appropriate points?

(4) How should the application of Sidner's rules to different anaphors in the current sentence be coordinated? How can the results of one application be used to help another, and how can mutually incompatible predictions by different applications be dealt with?

Each of the sections 6.1 to 6.4 in this chapter is devoted to answering one of these questions. Substantial extensions to Sidner's framework are described. Briefly, the answers I will argue for are as follows.

(1) The processes and results of anaphor resolution can, according to a "principle of anaphoric success" which will be formulated, be used to resolve many of the word-sense and structural ambiguities which depend on textual context for their resolution; that is, anaphor resolution often pre-empts word sense and structural ambiguity resolution. This fact largely determines SPAR's global organisation.

(2) Sidner's framework, in which local focus controls the resolution process, consulting other knowledge sources when necessary, is roughly correct as far as it goes. However, CSI (corresponding to Sidner's "special mode inference") should be invoked, if at all, only at the last possible moment. Furthermore, if, as in SPAR, no global focus mechanism exists alongside Sidner's local focussing algorithms, then certain changes to those algorithms are needed. Other changes are dictated by the need, in a shallow processing approach, to avoid using world knowledge where possible. Independently of global focus and shallow processing considerations, Sidner's framework must also be revised in the light of solutions to problems (3) and (4).

(3) Extending Sidner's PI rules to consider ISCs is best done not by adding new rules but by temporarily augmenting the focus registers with elements specified in the current sentence.

(4) Where the PI rules alone (i.e. without CSI) are unable to decide between two candidates for an anaphor, constraints arising from applying the rules to another anaphor often come to the rescue. Cases of complete incompatibility between the results of applying the rules to two different anaphors are fairly rare, but when they occur, the rules must be re-applied. In section 6.4 we will also develop a number of motivated preference criteria for dealing with anaphors which CSI, because of its limited power in the shallow processing framework, is unable to resolve.

Of the developments to Sidner's framework reported in this chapter, only the way in which candidate specifications are assessed (part of point (2)) and the introduction of collective preference criteria (part of point (4)) result from the use of a shallow processing methodology. The shallow processing character of the system is thus more apparent in its meaning representation and its CSI component than in its overall control structure.

6.1 Anaphoric, word-sense and structural ambiguity

6.1.1 Ambiguity resolution in textual context

Any ambiguities in the interpretation of an isolated written sentence should ordinarily be resolvable when the sentence is interpreted in its textual context; if not, the text is in some sense ill-formed. Context can in fact act in many different ways to resolve ambiguities; perhaps the most obvious example is where the antecedent of a pronoun appears in an earlier sentence. Since this research is concerned mainly with anaphoric ambiguity, we will limit our coverage of the influence of context on resolving word-sense and structural ambiguity to considering the cases where that influence can be seen as being mediated by anaphor resolution: that is, where the resolution of an anaphor leads naturally to the resolution of another ambiguity.

Crain and Steedman [1985] develop a psychological model of parsing, for which they hypothesise three principles of differing generality characterising the way in which context influences ambiguity resolution. Although Crain and Steedman discuss only structural ambiguity, these principles appear to be applicable to word sense ambiguity as well. From the most specific to the most general, they are:

> "The principle of referential success: If there is a reading which 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."

> "The principle of parsimony: If there is a reading that carries fewer unsatisfied but consistent presuppositions or entailments than any other, then, other criteria of plausibility being equal, that reading will be adopted as the most plausible by the hearer, and the presuppositions in question will be incorporated in his or her model."

"The principle of a priori plausibility: If a reading is more plausible in terms either of general knowledge about the world, or of specific knowledge about the universe of discourse, then, other things being equal, it will be favoured over one that is not."

It would seem that the cases of word-sense and structural ambiguity in which the influence of context can be seen as being mediated by anaphor resolution include all those to which Crain and Steedman's principle of referential success applies. For example, in each of

(6-1a) [1] John put a bowl on the floor.
[2] He picked up some biscuits.
[3] He put the biscuits in the bowl on the floor.

(6-1b) [1] John picked up some biscuits.
[2] He put them in a bowl.
[3] He put the biscuits in the bowl on the floor.

the preferred attachment of the PP "on the floor" is the one which allows more NP's to be resolved as anaphors.[1]

But since, given our definition of anaphora, not every anaphor specifies an existing item in the text model (e.g. consider substitutes (2.2.2) or "inferred specification" (5.4.2)) the anaphora-mediated cases go beyond those covered by Crain and Steedman's first principle. For example, in (6-2a) and (6-2b) below, the word sense ambiguity of "jack" disappears when the anaphoric NP "the jack" is resolved, even though neither first sentence implies the existence of a jack (either in the hearer's mental model or in reality):

(6-2a) [1] John put the cards on the table.
[2] He picked up the jack.

(6-2b) [1] John put the tools on the floor.
[2] He picked up the jack.

Furthermore, other things being equal, readings accessing more focussed antecedents tend to be preferred; in

(6-3) [1] John put the cards on the table.
[2] He put the tools on the floor.
[3] He picked up the jack.

we interpret "the jack" as one of the tools, not one of the cards.

SPAR is therefore intended to handle the range of contextual influences on word-sense and structural ambiguity resolution covered by the following variant of the principle of referential success:

The principle of anaphoric success: If there is a reading which succeeds in establishing some anaphoric relation with an element already present in the hearer's text model, then it is favoured over one that does not. Readings establishing anaphoric relations with

---

[1]Most of the example texts in this chapter and chapters 8 and 9 are ones that SPAR processes correctly, as shown in appendices A and B. Their somewhat pedestrian quality is due to practical considerations, notably the need to conform to Boguraev's analyser's grammatical coverage.

more focussed elements are favoured over less focussed.

Of course this principle is not the only one influencing SPAR's choice of reading. As we saw in 4.3.3, SPAR also takes account of the semantic density of each reading; and in any case, Boguraev's analyser will only have constructed readings in which most of the semantic preferences of verbs, prepositions and other words are satisfied.

6.1.2 SPAR's processing strategy

SPAR's processing strategy, involving the application of the principle of anaphoric success, is as follows. Each reading, represented by a current fragment of the TMN, is processed separately, and effectively in parallel. A score is assigned to each reading, partly on the basis of the intrinsic semantic density calculated during current fragment construction, but also, in accordance with the principle of anaphoric success, on the basis of the ease with which anaphors can be resolved: a penalty is incurred for each suggestion of the Sidner-derived anaphor resolution (AR) rules which is deemed unacceptable, and for each potential anaphor which cannot be resolved at all. When all anaphors have been resolved, the reading with the highest score is accepted.

For reasons which will be given later in this chapter, CSI is invoked, if at all, only after the AR rules have been applied to all potential anaphors. CSI is only invoked if some ambiguity remains: that is, if one or more anaphors are not fully resolved (because the AR rules required CSI to choose between two equally focussed candidates), and/or if more than one reading exists. The CSI mechanism's success in completing inference chains for a reading affects that reading's score. Experience suggests that it is safe to discard all but the highest-scoring readings after applying the AR rules and before CSI; this has the advantage of reducing the number of occasions on which CSI is needed.

Thus SPAR's overall processing strategy, in which anaphor resolution is embedded, is as follows.

Set the TMN and the focus registers to NIL.

For each sentence in the story in turn:

Call the fragment constructor to derive (typically) one current fragment from each dependency structure for the sentence.

For each reading (as represented by a current fragment):

Set the reading's score to its semantic density.

Apply the AR rules to each potential anaphor, using only linguistic knowledge (i.e. the structure matcher) to assess their suggestions. Subtract points from the score according to the number of rejected suggestions and the number of potential anaphors for which no suggested candidates were acceptable.

Apply configurational constraints.

Reject all but the highest scoring reading(s).

If more than one reading remains, or only one reading remains but there are anaphors for which several candidates remain, then for each reading:

> *Invoke CSI, and adjust the score according to the number of chains completed.*
>
> *If necessary, invoke some "collective" preference criteria* (to be described).

Accept the highest-scoring reading and merge it into the context according to the results of anaphor resolution. If there is a tie for highest score, apply some weak heuristics to select the winning reading.

These tie-breaking weak heuristics in fact involve comparing not current fragments but the dependency structures from which they were constructed.[2] They are:

> (1) If two structures differ in the sense they select for a word, the one with the smaller sense number is preferred: e.g. TOOL1 (a tool for mending things) is preferred to TOOL3 (a person being manipulated by someone in power). Dictionary entries for a word are assumed to be ordered with the most commonly used sense (in a neutral context) first; the fact that the story context is not necessarily neutral is ignored.
>
> (2) If two structures differ in the level at which they attach a case (typically originating from a PP), attachment to a verb phrase (i.e. a "clause" structure) is preferred to attachment to a noun phrase (i.e. a "noun-arg"); and, secondarily, low attachment is preferred to high. These attachment preferences approximate to Frazier and Fodor's [1979] rules of right association and minimal attachment.[3]

These heuristics, although better than nothing, are far from adequate. Some ways in which the system's reliance on them might be reduced are discussed in chapter 10.

---

[2]The comparisons are actually *made* before current fragment construction, but the results are only *used*, if at all, in the last stages of processing a sentence.

[3]Several authors have pointed out problems with Frazier and Fodor's rules; see Wilks et al [1985] for a review. Counterexamples exist where lexical and pragmatic factors override any syntactic preferences. Accordingly, in SPAR, the Frazier and Fodor rules are only applied when neither lexical preferences (i.e. semantic density) nor pragmatic ones (i.e. anaphor-mediated contextual influences) are decisive.

6.1.3 An example

The way in which anaphor resolution also results in non-anaphoric ambiguity being resolved, via the principle of anaphoric success, is illustrated by SPAR's processing of the text

(6-4) John put the dog on the table. He examined its legs.

After processing the first sentence, which has only one reading, the dog is established as discourse focus, with the table as a potential discourse focus. However, as the analyser dictionary contains two definitions of the noun "leg",

LEG1:	(THIS (*HUM PART))	("a distinct part of a human or animal")
LEG2:	(THIS (FURNITURE1 PART))	("a distinct part of a piece of furniture")

two readings are produced for the second sentence, differing in the sense of "leg" selected. When TMN fragments are produced for each reading, the *HUM and FURNITURE1 WSN nodes are, respectively, associated with "it", resulting in the assertions

(Specialisation: ITO-1 of *HUM)	for the LEG1 reading, and
(Specialisation: ITO-2 of FURNITURE1)	for the LEG2 reading.

The resolution of the anaphor "he" is straightforward. However, anaphor resolution for "its" proceeds as follows. For the LEG1 reading, SPAR's report on its progress (slightly abbreviated) is:

```
Resolving ITO-1 as POSSESSIVE pronoun
Applying df/af ambiguity rule
DOG1-1 yields successful DIRECT match
```

That is, the first suggestion of the AR rules, the TMN element for the dog, succeeds. For the LEG2 reading, on the other hand, the FURNITURE1 restriction prevents "it" matching with "dog", and we get

```
Resolving ITO-2 as POSSESSIVE pronoun
Applying df/af ambiguity rule
DOG1-1 yields no DIRECT matches
Penalty point because DOG1-1 failed
Predicting discourse focus: (DOG1-1)
DOG1-1 has already been suggested
Predicting potential discourse foci: (TABLE1-1 PUT1-1)
TABLE1-1 yields successful DIRECT match
PUT1-1 yields no DIRECT matches
```

The failed prediction of DOG1-1 resulted in a penalty point being imposed on the LEG2 reading. Thus the LEG1 reading is accepted because it allows "its" to be given a more focussed specification.

6.2 The anaphor resolution rules

As indicated earlier, the AR (anaphor resolution) rules used in SPAR are identical to Sidner's FDNP (full definite noun phrase) and pronoun interpretation rules in most respects except where coordination between knowledge sources, between sentential and contextual candidates and between anaphors is involved. Sidner's rules have been altered in SPAR only as required by the shallow processing methodology and in the few cases where they seemed obviously inappropriate. Although, as we saw in chapter 5, Sidner's rules are perhaps more complex than they need be, this research aims not to formulate a radically new set of rules but to explore within a shallow processing context the coordination issues that Sidner's rules, or any comparable set of local-focus-based rules, raise.

This section describes SPAR's rules for full noun phrases and pronouns, and justifies the use of linguistic knowledge alone to assess single suggested antecedents; coordination issues are dealt with more fully later in the chapter. Some of Sidner's rules have not been implemented in SPAR because the constructions they deal with occur only rarely in the type of stories SPAR processes, while on the other hand the rules have been tentatively extended to deal with indefinite noun phrases.

6.2.1 Rules for full noun phrases

SPAR's rules for full noun phrases follow Sidner's quite closely. Of the six specification relations listed in 5.4.2, only computed specification (necessary for NP's with ordinal modifiers) is not implemented. Cospecification (types (1) and (2)), associated specification and set-element specification are implemented much as recommended by Sidner. Inferred specification, however, is implemented rather differently. In a shallow processing system we wish to avoid CSI when possible, and in any case Wilksian CSI is geared towards constructing causal chains rather than the associations necessary to detect inferred specification. In SPAR, therefore, inferred specification is tackled by relaxing the information constraint (defined in 2.2.3). Such relaxation is in many cases equivalent to making the required assumptions (non-analytic inferences), as the following example shows.

Sidner [1979] exemplifies inferred specification using the text (repeated from 2.2.3)

(6-5) [1] The heiress lived the life of a recluse.
[2] She died under mysterious circumstances,
[3] but <u>the murderer</u> was never found.

Interpreting "the murderer" as "the murderer of the heiress" involves making the non-analytic inference that the heiress died by being murdered. However, linguistic knowledge is sufficient to make this inference. In SPAR, the formula for "murderer", and the formula for "kill" on which it depends, are

```
MURDERER1:    ((((NOTSAME MAN) OBJE) KILL1) (SUBJ MAN))))
KILL1:        ((*HUM SUBJ) ((*HUM OBJE) (DIE1 CAUSE)))
```

so that processing the phrase "the murderer" results in an implicit DIE1 node, with an unknown subject, being created in the TMN. The phrase can then be resolved by identifying the DIE1 TMN node with that for the "die" event in the second sentence. Making such an identification involves relaxing the information constraint because although the phrase "the murderer" presupposes some "die" event, the converse is not true.

The output produced by SPAR while processing (6-5) (slightly simplified for reasons irrelevant to our concerns here) is as follows. For each sentence, the input is given first, followed, after anaphor resolution, by SPAR's paraphrase of it.[4]

(6-6) [1]. THE HEIRESS LIVED AS A RECLUSE.
AN HEIRESS RESIDED LIKE A RECLUSE.

[2]. SHE DIED UNDER MYSTERIOUS CIRCUMSTANCES.
THE HEIRESS DIED UNDER MYSTERIOUS CIRCUMSTANCES.

[3]. THE MURDERER WAS NOT FOUND.
THE PERSON WHO KILLED THE HEIRESS WAS NOT FOUND.

The information constraint may need to be relaxed not only to resolve (collocative) anaphors like "the murderer" which specify a previously unmentioned entity; sometimes a (reiterative) anaphor specifies an already-mentioned entity but introduces new information about that entity. In such cases we must relax the information constraint and therefore make an assumption; the assumption is that any modifiers which cannot be accounted for as "given" information are in fact new. We will therefore differ from Sidner in extending the term "inferred specification" to cover these cases as well as cases of the "died...murderer" type. In

(6-7) The heiress lived as a recluse. Nobody saw the old woman.

the modifier "old" must be recognised as new information if the cospecification of "the old woman" with "the heiress" is to be established. However, if the head noun itself contains new information, the text reads more awkwardly and cospecification is perhaps less plausible:

(6-8) The old woman lived as a recluse. Nobody saw the heiress.

SPAR in fact attempts to handle three forms of inferred specification: where

(1) Anaphor and antecedent cospecify, but the anaphor has a modifier or modifiers that break the information constraint. Example: (6-7).

(2) The anaphor specifies an implicit element derived from the antecedent, and breaks the information constraint. Example:

(6-9) John was driving along the motorway. His Jaguar broke down.

where an implicit "vehicle" node is derived from "driving".

(3) An implicit element derived from the anaphor is identified with the element specified by the antecedent; the implicit element may contain more information than the antecedent element. Example: (6-6).

But in cases (2) or (3), the information constraint cannot be broken in arbitrary ways, or spurious identifications will be made. The constraint can only be broken if the word senses of the identified elements are sufficiently close. SPAR regards two word senses as "sufficiently close" if they are in an ancestor-descendant relationship in the WSN Specialisation: hierarchy and

[4]For an explanation of the perhaps surprising use of "an" in [1] here rather than "the", and of SPAR's paraphrase mechanism in general, see chapter 9.

neither word sense is a primitive. (We saw in 4.3.1 that the "word sense" of an implicit node is often a primitive). Primitives are ruled out because they are so general that matches between them do not constitute enough evidence to justify merging the two nodes. For example, when SPAR attempts to resolve "the murderer" in (6-6), the AR rules first suggest that the implicit node for the person murdered should be identified with the heiress. This is initially rejected on the grounds of the excessive generality of the "person" node; evidence that this behaviour is correct is provided by the awkwardness of (6-6) with the second sentence removed. The rules then go on to suggest that the implicit "die" node should be identified with the explicit specification of the "die" event in sentence [2]. This suggestion, which in fact entails the first one, is accepted.

The successful treatment of inferred specification in terms of the information constraint and linguistic knowledge, rather than by full CSI, is further evidence of the feasibility of a shallow processing approach. Other, less important differences between SPAR's full NP rules and Sidner's may be summarised as follows.

(1) "Usage ambiguity", between specific and generic readings of a noun phrase, is not considered, since it is rare in simple stories.

(2) The rules are applied to indefinite NPs as well as definite. Although cospecification is not normally a possibility for indefinites, other relations are. For example, in

(6-10) John entered a restaurant. <u>A waiter</u> came towards him.

the underlined phrase can be understood as having "a restaurant" as its antecedent, just as if it had been "the waiter".

(3) It is possible to use FDNPs to refer to entities that have drifted out of focus and are therefore never considered by Sidner's rules. SPAR therefore searches, if comparison with focussed elements does not resolve an FDNP, for any out-of-focus elements introduced earlier in the text with which cospecification is possible without breaking the information constraint. This search is carried out before inferred specification is considered; it is a simple but, given SPAR's aims, adequate substitute for a global focussing mechanism.

6.2.2 Rules for pronouns

Except where considerations of intrasentential candidate antecedents (discussed in section 6.3 below) are concerned, SPAR's AR rules for definite pronouns differ in only minor respects from Sidner's PI rules. Other than in the elimination of Sidner's recency rule, which as we saw in 5.4.3 is usually incorrect, the only difference is that for non-agent pronouns, nodes in the discourse focus stack are suggested if foci and potential foci are rejected. This brings the rules for non-agents into line with those for agents, for which, in Sidner's algorithms, the actor focus stack is suggested when necessary.

Sidner's "co-present foci" rules for noun phrases of the form "this ...", "that ...", "(the) one" and "(the) other" are not implemented because such phrases were not encountered in the stories processed. However, "one" phrases (both with and without modifiers) are catered for by SPAR as follows.

SPAR's pronoun rules are applied not only to definite pronouns but also to descriptional anaphors (i.e. indefinite pronouns and indefinite and definite noun phrases headed by indefinite pronouns). However, a wider range of relationships between anaphor and antecedent is assessed than just cospecification. The algorithm used for establishing the relationship between such an anaphor and a suggested antecedent is:

(1) If the anaphor is definite, apply the structure matcher to see if anaphor and candidate antecedent may cospecify without breaking the information constraint. If they may, decide whether the anaphor is contrastive: ensure that there are one or more elements in the TMN that have the same (or a subordinate) head word sense as the candidate and that the anaphor cannot match any of them without breaking the information constraint. If no contrast is found, the anaphor is odd. Examples:

(6-11a) John picked a red flower and a yellow flower. Mary asked him for the red one.

(6-11b) ℵ John picked a red flower. Mary asked him for the red one.

(2) If the candidate is plural, hypothesise that the anaphor specifies a member (or, if plural, a subset) of the set specified by the candidate. This can occur if the head word senses of the anaphor and candidate match, and the anaphor has an additional modifier and/or is indefinite. (This is an extension of Sidner's "set-element specification"). Example:

(6-12) John picked some flowers. Mary asked him for (the red one / a red one / one).

(3) Hypothesise that the anaphor and antecedent are related only at the level of sense: i.e. that the anaphor inherits the head word sense of the antecedent, but that their specifications are not related in any other way. Example:

(6-13) John liked bananas. He asked Mary for one.

When several candidates are suggested at once, tests (1) to (3) in turn are applied to all of them together, and the first candidate to satisfy a test is accepted. Initial indications are that this approach leads to correct results; however, it has not been thoroughly tested.

Indefinite pronouns in phrases of the form "<IndefinitePronoun> of <NP>" are not dealt with by the above mechanism; instead, they are treated as parasitic. Thus in

(6-14) Wendy examined the T-shirts on the table. She picked one of them up.

the "them" is resolved as a normal definite pronoun, and the "one" is then assumed to specify a member of whatever set is specified by "them".

SPAR's treatment of plural definite pronouns removes some indeterminacy in Sidner's rules. In section 2.2.1 we saw that a plural pronoun may cospecify several items in the text that have not been mentioned together in syntactic construction. For plural pronouns, Sidner's rules at various points make suggestions such as "predict from DF and PDF together". However, this does not mean that the suggested specification is necessarily composed of *all* the DF and PDF items. When resolving the pronoun in

(6-15) John met Mary in the park. A policeman saw them,

the aggregate of the DF and PDF is John, Mary, the park and the meeting event. All of these can be seen by a policeman, yet we understand "them" as meaning only "John and Mary". Thus as well as the semantic preferences imposed by the pronoun and the role(s) it fills, there seems to be a tendency to construct "group" specifications from entities of the same semantic category (in some broad sense). In SPAR this tendency is encoded as a stipulation that the members of a group must have the same head primitive. Since in the first sentence of (6-15) only John and Mary have the same head primitive (MAN, whereas those of "park" and "meet" are SPREAD and SENSE respectively), "them" is understood as John and Mary. The rule that head primitives of group members must be the same works well, although in some cases it fails, for example in

(6-16) John took the dog for a walk in the park. A policeman saw them.

where "John" has head primitive MAN and "dog" has head primitive BEAST. (Although both MAN and BEAST are subsumed by the starred primitive *ANI, starred primitives are in most cases too general to give sufficient discrimination, even though in this particular case their use in place of ordinary primitives would give the right answer).

6.2.3 Assessing candidate antecedents

In Sidner's framework, inference is invoked both in "normal mode", to decide whether accepting a single suggested antecedent for an anaphor would lead to a contradiction, and in "special mode", to decide between two candidates judged equally focussed by the PI rules. Although Sidner recognises that the inference mechanisms for the two modes may be quite different, she does not attempt to state in detail what inference mechanisms and types of knowledge (e.g. linguistic or non-linguistic, analytically true or uncertain) may be appropriate for, on the one hand, detecting a contradiction (that is, complete implausibility), and on the other hand, choosing between candidates.[5] However, it seems clear that a genuine contradiction can be established only by the use of definitional (linguistic) knowledge and analytically true inferences; if any non-analytic, uncertain inferences are made, then an apparent contradiction may be due to those inferences being incorrect.

Because SPAR uses preference semantics, it is necessary to decide which of the criteria Wilks used to resolve pronouns qualify as purely linguistic and/or analytic. Only these criteria should be used in Sidner's normal mode. For

---

[5]Clearly normal mode must be a component of special mode in the sense that if the PI rules present two candidates and demand special mode inference, the special mode inference mechanism should first invoke normal mode inference on each one and only proceed further if they are both approved. The issue here is, rather, whether some types of reasoning and knowledge appropriate for special mode inference proper are inappropriate for normal mode.

present purposes, Wilks' pronoun resolution strategy, which was introduced in chapter 1, can be summarised as follows:

> Follow the steps below until only one candidate survives.
>
> (1) Collect candidates and match them with the pronoun.
>
> (2) Apply preference restrictions to case fillers: e.g. "drink" prefers a liquid object.
>
> (3) "Extract" new templates from old using analytic inference rules, and try to match new templates to bind the pronoun.
>
> (4) Apply non-analytic CSI rules to derive new templates, and try to match them to bind the pronoun.

These four stages represent a progression from very strong, analytic criteria to weaker, non-analytic ones. Stages (1) and (2) are purely linguistic, and can therefore safely be used in normal mode inference, while stage (4) is non-analytic and therefore unsafe. Since extractions (stage (3)) represent analytically true consequences of existing facts, it might be thought that they too are appropriate for normal mode. However, closer examination reveals that although *making* the extractions is analytic, *matching* them with one another to detect contradictions would not be. For example, for the text

(6-17) [1] Bill stayed at home.
[2] John went to London.
[3] Then he travelled from Birmingham to Cambridge,

Sidner's PI rules would first suggest John as the referent of "he". If normal mode inference involved the search for contradictory chains composed of extractions, the system would correctly deduce that after [2] John was definitely in London, and that before [3] John (temporarily identified with "he") was definitely in Birmingham. Since being in Birmingham is incompatible with being in London, a contradiction would be detected when these two extractions were matched. John would be rejected as referent, and the PI rules would suggest Bill instead. Since Bill's home could be in Birmingham, no contradiction would result, and Bill would be accepted.

However the more natural interpretation of (6-17) is that between events [2] and [3], John travelled from London to Birmingham. In other words, the focus preference of John over Bill is strong enough to resist any apparent contradictions based on extractions, because completing an inference chain can involve making an uncertain assumption (in this case, that John stayed put between [2] and [3]). Thus the use of extractions is inappropriate in normal mode inference, and is confined in SPAR to special mode inference.

In normal mode inference, therefore, SPAR uses only knowledge corresponding to Wilks' criteria (1) and (2) in deciding whether to reject candidates. Candidates are assessed using the structure matcher described in chapter 4. Case filler preferences detected by criterion (2) have already been applied during construction of the current fragment. Such preferences are important in restricting the range of possible specifications for very general pronouns like "it" and "they". For example in

(6-18) I bought the wine. I sat on a rock. I drank it,

SPAR correctly rejects the AR rules' suggestion of the rock as the referent of "it" because the "drink" formula restricts "it" to matching a liquid. The subsequent suggestion of the wine is, however, accepted.

But there are situations where a suggested antecedent leads not to a logical contradiction but to an interpretation so bizarre that it should be rejected. SPAR's structure matcher cannot detect such cases, and is therefore occasionally too permissive in approving candidate antecedents. However the fact that the structure matcher judges a candidate antecedent acceptable does not always mean that that antecedent will ultimately be accepted. We will see in section 6.4 that consideration of sentential candidate antecedents and of the interactions between anaphors can result in the AR rules' conclusions being revised or refined. For example in

(6-19) I took my dog to the vet on Friday. He bit him in the hand.

the structure matcher does not detect the oddness of accepting the vet as referent of "he"; it only knows that animate entities can bite, and not that it would be bizarre for a vet to bite his patients. (In fact, since "him" may not have been resolved, the system might not be able to detect the oddness even if it knew that vets do not bite patients). However, the suggestion that "he" specifies the vet is ultimately rejected because "him" can only specify the vet (since, according to the formula for "hand", only humans have hands) and the MGC (minimal governing category) configurational constraint states that "he" and "him" cannot cospecify. SPAR's processing of this text will be described more fully in section 6.4.

Thus the use of linguistic knowledge alone for normal mode inference in SPAR is justified quite independently of the shallow processing hypothesis. In a shallow processing approach we want to avoid using world knowledge whenever it is safe to do so; but it turns out that not only is it safe to ignore world knowledge in normal mode inference, it is actually not safe to do anything else.

Common sense inference is, however, clearly necessary for special mode inference. In SPAR, when the pronoun rules demand special mode inference, CSI is not invoked immediately; instead, the alternative predictions involved are all returned as the result of the pronoun rule application. This is necessary because, as section 6.4 will show, CSI and the other criteria relevant to choosing between alternative predictions are best applied to the sentence as a whole rather than to each individual anaphor in it. SPAR thus invokes CSI at most once per reading of a sentence, and postpones invocation as long as possible in the hope that applying other, linguistic criteria will make it unnecessary.

6.3 Intrasentential candidate antecedents

As we saw in the last chapter, Sidner's PI rules do not deal with intrasentential candidate (ISC) antecedents for pronouns. However, sentential cospecification is very common in most kinds of text, including simple stories, and so Sidner's framework must be extended in SPAR to deal with it. In this section, I will argue that this should be done not by inserting extra suggestions in the PI rules but by temporarily augmenting the contents of the focus registers. First, however, a potentially serious ordering problem must be disposed of.

6.3.1 When an antecedent is important

In chapter 2 we defined an antecedent as a phrase whose meaning must be taken into account to correctly interpret an anaphor with which it has a cohesive relationship. Therefore in

(6-20) [1] John bought <u>a new car</u> the other day.
[2] He also bought a television.
[3] <u>The car</u> was so unreliable that he soon wished he'd never set eyes on <u>it</u>.

the phrase "a new car" is the antecedent of "the car", and both of the phrases "a new car" and "the car" are antecedents of "it", because knowing their meaning (John's new car) is essential to determining the meaning of "it".[6] Analogous observations hold for the underlined phrases in

(6-21) [1] John bought a new car the other day.
[2] He also bought <u>a television</u>.
[3] <u>It</u> was so unreliable that he soon wished he'd never set eyes on <u>it</u>.

The fact that an anaphor may itself be the antecedent of another anaphor in the same sentence seems at first to present a serious coordination problem. How can we evaluate the plausibility of a candidate antecedent until we know what the candidate itself specifies; or, alternatively, how can we decide in advance in what order to resolve the anaphors in a sentence so that we are never faced with an unresolved candidate antecedent? Linear ordering is inadequate because of the possibility of cataphora; however, some solution to this problem is essential for the proper treatment of intrasentential anaphora.

My answer is that when the AR rules suggest candidates for assessment, certain kinds of intrasentential antecedence can be ignored, and that consequently there is an ordering for anaphor resolution which effectively guarantees that, except for certain, hopefully rare, cases to be discussed, unresolved candidate antecedents never arise.

To see this, note that there is an important difference between the texts (6-20) and (6-21). In both, after [2], the television is more focussed than the car. In (6-20), the phrase "the car" in [3] brings the car back into focus so that "it" is able to specify it. In (6-21), the first "it" has little or no effect on the state of focus, since the television is already strongly focussed; the fact that the first "it" is interpreted as specifying the television is therefore largely irrelevant, in focussing terms, to the fact that the second "it" is also so interpreted. These claims can be verified by observing what happens to the interpretation of the (final) "it" when the words "(the car/it) was so unreliable that" are deleted from each text. In (6-20), the interpretation switches from the car to the television, whereas in (6-21) it remains unchanged.

---

[6]It is not claimed that both ***antecedents*** must be considered when "it" is being resolved; merely that their (shared) ***meaning*** must be considered. Nevertheless, both phrases are antecedents according to our cohesion-based definition.

In extending Sidner's rules to deal with intrasentential antecedents, we therefore need to take account of cases like (6-20) but not cases like (6-21). That is, considering for the moment only definite anaphors and not descriptional ones, an intrasentential antecedent is only important (a) if it introduces a new element, or (b) if it makes an existing but not very focussed (or unfocussed) element more focussed so that the anaphor is able to specify it where it would not otherwise be able to. But case (b) can in fact only occur when the antecedent is more informative than the anaphor: i.e. in texts like (6-20) but not in texts like (6-21).

Thus since definite pronouns are minimally informative and never introduce new elements,[7] the only cases of possible intrasentential antecedence we need to consider are

(1) where neither the anaphor nor the possible antecedent is a definite pronoun, and the antecedent is strictly more informative; and

(2) where the anaphor is a definite pronoun and the possible antecedent is not.

Therefore, if we always resolve more informative non-pronominal anaphors before less informative, and full NPs before definite pronouns, any potentially important ISCs (intrasentential candidates) will already have been resolved by the time they are considered. SPAR's structure matcher (see 4.3.2) can determine whether one anaphor is more informative than another by matching their TMN nodes with the information constraint switched on.

How does descriptional anaphora (in Halliday and Hasan's terms, nominal substitution or ellipsis) fit into this picture? On the one hand, a descriptional anaphor (e.g. "one" or "a red one") can introduce or make focussed a new entity which acts as the antecedent of a definite pronoun, either anaphorically or cataphorically:

(6-23a) John sold his old car. He bought <u>a new one</u> and drove <u>it</u> away.
(6-23b) John has two cars. When he bought <u>it</u>, <u>the newer one</u> was faster.

On the other hand, such a phrase can itself be either anaphoric or cataphoric (at the level of sense rather than specification):

(6-24a) Before I bought <u>a new car</u>, I never realised how noisy <u>my old one</u> was.
(6-24b) Before I bought <u>a new one</u>, I never realised how noisy <u>my old car</u> was.

It is thus necessary to resolve such anaphors after full NPs and before definite pronouns. The ordering adopted in SPAR is therefore the following:

---

[7]Definite pronouns occasionally ***mention*** an element for the first time, as in

(6-22) I went to a concert last night. <u>They</u> played Beethoven's Fifth.

However, such elements are not obviously ***new***; the existence of the orchestra is quite strongly implied by the use of "concert".

(1) Full NPs, with more informative before less informative, and otherwise in textual order;

(2) Descriptional anaphors;

(3) Definite pronouns.

An anaphor (in the strict, i.e. non-cataphoric sense) relating to its antecedent by associated or inferred specification (see 5.4.2) will be correctly dealt with by this ordering. However problems will arise for associated or inferred cataphora, as in

(6-25) Someone has removed the indicators from my car.

(where the cataphor "the indicators" relates to "my car" by inferred specification), if the car has not been previously mentioned. No such cataphors arise in the texts SPAR has processed. To deal with them properly, the system could perhaps include some tentative intrasentential anaphor (and cataphor) resolution during TMN construction. However, how the results of such processing could be integrated with the results of the subsequent main anaphor resolution phase remains to be worked out.

6.3.2 Two ways to extend Sidner's algorithms

How should consideration of ISCs be incorporated into Sidner's framework? Perhaps the most obvious possibility is to insert extra rules; thus if originally the PI rules predicted the DF (discourse focus), and then the PDFs (potential discourse foci), we might enlarge the rule set to predict ISCs in between these two. However, there are at least two reasons why this is a bad idea.

The first reason is that much of the complexity in Sidner's rules is due to the existence of two independent types of focus register (discourse and actor); so introducing a third independent source of predictions (the intrasentential one) would, if thoroughly carried through, make the complexity far worse.

The second reason is that focus preferences between contextual and intrasentential candidates are, as we will see in more detail below, very hard to establish, and so extra rules would be difficult to formulate. When resolving a non-agent pronoun, Sidner's rules strongly prefer the DF to PDFs: that is, PDFs are only considered if normal mode inference decides that the DF is implausible. However, it would not be safe for the rules to prefer the DF over ISCs (or vice versa) in the same strong way. If such preferences exist, they are weak (i.e. secondary to the results of special mode inference - see 5.4.3); thus the first rule for non-agent pronouns would have to say something like "suggest the DF and (some) ISCs together, performing special mode inference if more than one is acceptable, and preferring the DF only if special mode inference is inconclusive".

Since we are forced to consider contextual and intrasentential candidates in the same rule, an alternative approach suggests itself: to temporarily augment the focus registers themselves, and apply Sidner's rules with only the few necessary modifications. The register contents can be augmented with ISCs so as to reflect any weak preferences (for example, we might add ISCs to the beginning of the PDF list rather than to the end), and the rules themselves can remain virtually unaltered. In this way, Sidner's framework can be extended to cover ISCs without any great increase in complexity.

6.3.3 Augmenting the focus registers

Given that Sidner's rules reflect weak and strong focus preferences between contextual candidates (i.e. the various actual, potential and stacked foci), what analogous preferences hold between different ISCs and between intrasentential and contextual candidates? In 5.3.2 we looked at the algorithms of Winograd, Hobbs, and Guenthner and Lehmann. All three algorithms preferred intrasentential to contextual candidates and (with minor exceptions in Hobbs' case) anaphora to cataphora, subjects to objects and objects to oblique cases. Conversely, Sidner leans towards favouring the DF (which is contextual) over ISCs.

None of these authors claim that their preference rules are always correct; and indeed, Cantrall [1975] found in a psycholinguistic experiment that "test subjects differed idiosyncratically in their tendencies to determine coreferents in each of the following ways, at least: grammatical function, relational function, order, distance, relative position in the utterance, and pitch, not to mention perception (and indeed creation) of 'semantic' associations". Thus there is unlikely to be any complete and correct set of preferences, whether expressible in Sidner's framework or not; and even if there is, finding them would be very difficult because of all the complicating factors. The best that can be hoped for is a set of preferences, like Hobbs', that works most of the time; and these preferences should be used only as a last resort. (Indeed, we will see in section 6.4 that it is usually possible to avoid using them).

But having said that, one strong preference does seem to exist. ISCs are strongly preferable to candidates not mentioned in either the current sentence or the previous sentence; that is, they should be predicted before stacked foci are considered. We therefore only need to consider what weak preferences hold between ISCs and current and potential foci (and of course other ISCs).

A useful starting point for establishing such preferences is Hobbs' algorithm, which is fully specified and has been shown to be quite accurate. This algorithm is used in SPAR to impose a weak ordering on ISCs. The way these ordered candidates are added to the focus registers is determined by procedures based on the following observations:

> (1) Since the default action in a text seems to be to maintain the focus rather than switch to a new one, the DF is preferred weakly to ISCs. Sidner [1979b] notes that the DF (Bruce) is preferred for "his" in the following:
>
> (6-26) [1] I want to have a meeting this week.
> [2] Bruce will be the guest lecturer.
> [3] He will speak on slavery in ant colonies.
> [4] Mike wants to read <u>his</u> report before the talk.
>
> But we can see that this preference is only weak by changing [4] to
>
> [4'] Mike wants to invite <u>his</u> friends to the talk.
>
> Here it more likely that Mike wants to invite his own friends than that he wants to invite Bruce's, but the latter is not impossible.

(2) On the other hand, an ISC occurring before the anaphor is marginally preferable to a potential focus. In

(6-27) [1] Mary arrived at the club with John.
[2] Susan was telling Bill's sister all about <u>his</u> behaviour the previous evening.

where John becomes both PAF (potential actor focus) and PDF after [1], there is perhaps a slight preference for Bill, rather than John, as referent of "his".

(3) However, cataphora (i.e. an ISC after the anaphor) is less plausible than cospecifying with a potential focus, at least in children's stories; consider for example

(6-28) [1] Mary arrived at the club with John.
[2] Susan was telling <u>his</u> sister all about Bill's behaviour the previous evening.

where this time John is the weakly preferred referent of "his".

Thus if strong preference is marked by ">>" and weak preference by ">", the situation may be summarised as follows. Without ISCs, Sidner's rules (for "normal" pronouns) essentially stipulate the preferences

(6-29) DF >> PDF >> DF stack

However, observations (1) to (3) suggest that when ISCs are considered, the preferences become

(6-30) DF > pre-anaphor ISC > PDF > post-anaphor ISC >> DF stack

The examples given are far from conclusive, and there are other, stronger, relevant factors, some of which are analysed in section 6.4 below. However, in accordance with (6-30), SPAR's focus registers are temporarily augmented as follows just before the AR (anaphor resolution) rules are applied to a pronoun.

(1) Apply Hobbs' algorithm (slightly altered)[8] to derive an ordered list of ISCs.

(2) If the current pronoun could consistently (according to the structure matcher) specify the DF, then append those ISCs ***preceding*** the pronoun to the DF register (so we now may have several discourse foci). If the current pronoun cannot specify the DF, append preceding ISCs not to the DF, but onto the front of the PDF list. This apparently ad-hoc behaviour follows from the observations above that while the DF is strongly preferred to PDFs, the DF is only weakly preferred to pre-anaphor ISCs, which in turn are only weakly preferred to PDFs (see (6-29) and (6-30)).

---

[8]Hobbs' algorithm as published [1976] does not predict all possible cataphoric candidates; it ignores those which cannot be reached from the anaphor node without going down through an S or NP node. In [2] of (6-28) above, the "Bill" NP node can only be reached from "his" by passing down though the "Bill's behaviour" NP node that dominates it, and so is ignored. In SPAR, any cataphoric candidates not reached by Hobbs' algorithm are appended in textual order to the list of those reached.

In either case, append those preceding candidates which are animate to the actor focus register (so we may now have several actor foci).

(3) Append those ISCs *following* the pronoun to the PDF list and (when animate) to the PAF list.

When applied to "his" in (6-28) above, the results would be as follows. At the end of [1], we have

DF = Mary
AF = Mary
PDF = the club, John, the arrival
PAF = John

Hobbs' algorithm returns the candidates Susan, Bill's behaviour, the previous evening, and Bill, in that order. On applying steps (2) and (3) of the algorithm, we get

DF = Mary, Susan
AF = Mary, Susan
PDF = the club, John, the arrival, Bill's behaviour, the previous evening, Bill
PAF = John, Bill

Thus John precedes Bill in the PDF list, and is accordingly weakly preferred as referent of "his".

6.4 Interactions between anaphors

Many of the factors that influence anaphor interpretation are constraints or preferences not on the interpretation of single anaphors in isolation but on the relationships between their interpretations. A syntactic constraint such as Reinhart's c-command rule says nothing about what an individual anaphor may or may not specify; rather, it says that given pairs of noun phrases cannot cospecify. A completed Wilksian CSI chain typically binds several pronouns rather than one; accepting the chain as valid means we must accept all its bindings as correct. Factors such as c-command and CSI which apply to several anaphors at a time will be called <u>collective</u>; factors such as semantic compatibility and those aspects of focussing embodied in the AR rules will be called <u>distributive</u>.

SPAR uses distributive factors, encoded in the AR rules, to constrain the interpretation of each anaphor to a small number of alternatives; it then uses as many of the collective factors to be described in this section as are necessary to reduce the number of alternative interpretations per anaphor to one. When several alternative interpretations for an anaphor have been suggested by the AR rules and not yet eliminated, we will say that that anaphor is <u>undetermined</u>. The part of SPAR that integrates the predictions of the distributive AR rules with those of subsequently-applied collective factors is the <u>arbitrator</u>.

With one exception (syntactic constraints) collective factors are weaker than distributive ones, and therefore the arbitrator does not allow collective factors other than syntactic constraints to overturn the distributive factors' decisions altogether; they can only further constrain them.

The collective criteria SPAR uses are discussed in the order of their application: strongest to weakest. Thus subsection 6.4.1 concerns syntactic constraints, 6.4.2 describes the use (but not the operation) of CSI, and 6.4.3 discusses three weaker preference criteria.

6.4.1 Applying syntactic constraints

When the AR rules have terminated, the first collective factor invoked, in fact whether or not there are any undetermined anaphors, is that of syntactic constraints. Approximations to Reinhart's c-command and MGC constraints (see section 2.4) are applied to detect mutually inconsistent predictions for different anaphors. For the text

(6-31) I took my dog to the vet on Friday. He bit him in the hand.

the AR rules predict that "he" is either the dog or the vet, and "him" is the vet (since, according to the formula for "hand", only people have hands). The MGC rule tells SPAR that the predictions "he=vet" and "him=vet" are inconsistent, since the two pronouns have the same MGC (the topmost S node) and are non-reflexive. The arbitrator rejects the "he=vet" prediction because it has an alternative (he=dog) whereas "him=vet" does not. Similarly, for

(6-32) I took my dog to the vet on Friday. He injected him with a new medicine.

the AR rules predict that "he" is the vet (not the dog, since animals cannot perform injections) and "him" is either the dog or the vet. This time it is the "him=vet" prediction that arbitrator rules out on the basis of the MGC constraint. In these two texts, as in many others, syntactic constraints are sufficient for the arbitrator to determine all anaphors and therefore make CSI unnecessary.

As in our two examples, the arbitrator always displaces predictions with alternatives in favour of those without. If two clashing predictions both have alternatives, nothing is done for the time being, because neither is clearly wrong. Occasionally, however, neither prediction in a clash has an alternative, and more drastic action is required; the AR rules must have terminated too soon on at least one of the anaphors involved. In such cases, the AR rules are reapplied to those anaphors, starting from where they terminated before, in order to generate some further, alternative predictions. If this does not solve the problem, it is assumed that one or both of the anaphors is unresolvable, and the current reading is rejected. If all readings are rejected the sentence is ignored. (When all readings are rejected, it is usually because a wrong decision has been made earlier in the text; in chapter 10 I will suggest ways in which SPAR could be made to go back and correct such errors).

How can the c-command and MGC rules, which are both stated in terms of surface syntax, be applied accurately to pieces of the TMN? In practice it is usually possible to apply them because, as we saw in chapter 4, case relationships between explicit TMN nodes are nearly isomorphic to those in the analyser dependency structure from which they originated, and the dependency structure is sufficiently shallow for relevant aspects of the shape of the corresponding surface syntactic parse tree to be determined quite reliably. Almost every noun-arg in the dependency structure corresponds to an NP node, and almost every clause (other than adjectival relatives, which can be recognised) corresponds to an S node. Further, it is usually true that, for active voice sentences, an AGENT case link corresponds to a syntactic subject, OBJECT and RECIPIENT links to syntactic objects, and all other case links to noun-args to prepositional objects.

The relationship between dependency structures and surface syntactic trees is not always as straightforward as these assumptions suggest, and syntactic relationships can perhaps never be decided with complete certainty on the basis of one of Boguraev's dependency structures. However, no problems have been encountered in this area in the texts SPAR has processed, and in any case the difficulty is of no theoretical importance to the present work. Rather, it is an indication to builders of sentence analysers that if, as seems likely, constraints such as c-command are real and act at the level of surface syntax, then certain aspects of surface syntactic structure should be preserved in the sentence representation if anaphors may have to be resolved later.

6.4.2 Incorporating the results of common sense inference

Syntactic constraints are often sufficient to remove any indeterminacy in the results of the AR rules and make CSI (i.e. Sidner's special mode inference) unnecessary. However, if any anaphors remain undetermined, and/or if more than one sentence reading survives (i.e. if the processing performed up to this point has not revealed any differences in semantic density or ease of anaphor resolution), CSI is invoked. In the former case, any chains it completes will be a source of predictions which should enable anaphors to be fully determined; in the latter case, completed chains are some indication that the current reading fits in well with context and should be preferred over other readings.

The functioning of the CSI mechanism, and the way it is simplified and made more reliable by being constrained by focussing, will be described fully in chapter 8. For the moment, we are only concerned with the CSI mechanism's input and output and the way these relate to the overall anaphor resolution process.

When CSI is invoked, it is handed a set of undetermined anaphors (or, if inference is only taking place because of the existence of competing readings, the pronouns in the current sentence) and their possible specifications as determined by the AR rules. CSI returns an ordered list of predictions, each prediction resulting from a completed chain and consisting of a set of anaphor-candidate bindings. The application of CSI rules is constrained by the sets of anaphors and candidate specifications initially handed to the CSI mechanism, so that most bindings will involve only members of those sets.

The fact that a completed chain typically binds several anaphors to antecedents means that CSI is collective and not distributive. If CSI is treated distributively - that is, if the AR rules are able to invoke it to resolve a single pronoun - several problems arise. Suppose the current sentence is "He hit him", and the referents John and Bill are suggested for "he". Suppose further that the system knows that John hates James and can infer that hating someone can motivate hitting them. If CSI is invoked while "he" is being resolved, the system cannot decide whether to accept a chain binding "he" to John and "him" to James without knowing whether the AR rules will predict James as referent of "him": i.e. whether James is sufficiently in focus. Furthermore, invoking CSI separately for each anaphor can be inefficient, as the inferences made on each occasion may overlap significantly. These problems are avoided if CSI is only invoked after the AR rules have terminated on all anaphors.

Of the predictions returned by CSI, the arbitrator only accepts those consistent with the results of the AR rules. Thus if in a given text, the AR rules had decided that some pronoun "he" specified either John or Bill, then a CSI prediction binding "he" to Bill would be acceptable (as long as its other bindings were also satisfactory) but one binding "he" to Fred would not be. This filtering of CSI predictions takes place because CSI is used only to choose between candidates for which focus has no strong preferences, and not to generate suggestions itself. This limitation on the role of CSI, and the resultant easing of its task, is one of the main attractions of Sidner's theory.

When two CSI predictions are mutually inconsistent (e.g. if they bind the same pronoun to different antecedents or if together they would break a syntactic constraint), the arbitrator rejects the one constructed from a longer, and therefore less reliable, chain of inferences. If the chain lengths are the same, both predictions are provisionally accepted; the clash will be resolved by the "MCS" mechanism explained below.

The way the arbitrator combines the predictions of the AR rules, syntactic constraints and CSI is illustrated for the third sentence of the example text presented in chapter 1, repeated here in part:

```
(6-33)  [1]. JOHN PROMISED BILL THAT HE WOULD MEND HIS CAR.
             JOHN PROMISED BILL THAT JOHN WOULD REPAIR BILL'S CAR.

        [2]. HE TOOK IT TO HIS FRIEND'S GARAGE
             TO JOHN'S FRIEND'S GARAGE, JOHN CONVEYED THE CAR.

        [3]. HE TRIED TO PERSUADE HIS FRIEND THAT HE SHOULD LEND HIM SOME TOOLS.
             JOHN ATTEMPTED TO CONVINCE JOHN'S FRIEND THAT THAT FRIEND SHOULD LOAN
                JOHN SOME REPAIR IMPLEMENTS.
```

When [3] is processed, the AR rules find that both "he's" and the "him" are ambiguous between John and his friend; Bill is not considered since he is out of focus. The c-command constraint rules out the possibility of the first "he" cospecifying with "his friend"; the arbitrator therefore rejects this possibility and firmly resolves the first "he" as John. The MGC constraint prevents the second "he" and "him" from cospecifying. This fact is remembered, but since both pronouns are still ambiguous, the arbitrator cannot firmly resolve either of them yet. CSI is therefore invoked.

Rather than reasoning, as a more powerful inferencer might, that "him" is John because John is likely to want tools to mend the car, CSI simply predicts that "him" and the first "he" cospecify, using a shallower, more general rule stating that people are more likely to want to possess things themselves than to want other people to possess them. Since the arbitrator has already decided that the first "he" is John, it decides that "him" is also John and not the friend.

Now that "him" is firmly resolved as John, the arbitrator is able to apply the constraint that the second "he" and "him" (=John) do not cospecify. It rules out the possibility that the second "he" is John, and selects the friend as the referent.

In many cases, as in this one, the arbitrator is able to resolve any undetermined anaphors using only the predictions of the AR rules, syntactic constraints and possibly CSI. Any wrong predictions made by CSI will usually be rejected because they are inconsistent with other, more reliable predictions. However, the results of CSI are sometimes incomplete rather than (or as well as) wrong: that is, CSI can make errors of omission (failing to bind anaphors) as well as errors of commission (attempting to bind anaphors to the wrong candidates). When this happens, further processing, which will now be described, is necessary.

6.4.3 Some further collective criteria

The techniques discussed up to this point are all quite reliable in the sense that any answers they provide are, if accepted by the arbitrator, very likely to be correct. However, because of the limited power of the CSI mechanism, some anaphors remain undetermined even after CSI. What preference criteria can we use to make intelligent choices in such situations? Sidner's rules provide weak heuristics for this purpose, but we saw in section 6.3 that neither Sidner's nor anyone else's weak *distributive* heuristics are reliable enough to be used except as a last resort. It is therefore important to formulate, if possible, some *collective* criteria which, although less reliable than the factors we have looked at so far, are nevertheless stronger than weak distributive heuristics.

Once the arbitrator has applied the results of CSI to the set of existing predictions, hopefully reducing the size of that set, SPAR finds the <u>maximal consistent subsets</u> (MCSs) of the set, and uses up to three collective criteria (in addition to the configurational contraints and CSI already discussed) in succession[9] to choose between them. Thus if for the text

(6-34) John spoke to Bill. <u>He</u> hit <u>him</u>

CSI failed to complete any chains, the set of predictions would be (in simplified form)

{he=John, he=Bill, him=John, him=Bill}.

Because the MGC constraint prevents the two pronouns from cospecifying, the arbitrator finds two maximal consistent subsets:

(MCS1) {he=John, him=Bill}
(MCS2) {he=Bill, him=John}

The first criterion used to choose between MCS's is that of <u>repetition</u>. This can be viewed as an attempt, at a very shallow level, to detect certain kinds of coherence relation. For every explicit TMN node in the current fragment for which the AR rules made no suggestions (including clausal nodes, to which

---

[9] I have not experimented with any orderings of the criteria other than the ones described here. The order presented here is motivated by the apparent usefulness of the criteria on the stories SPAR has processed. However, in those stories it is unusual for more than one of the criteria to apply in a given situation; the ordering may therefore not be very important.

they were not applied), a fuzzy "repetition match" is carried out with explicit nodes in the "context" (non-current-fragment) part of the TMN. This match does not attempt to establish possible cospecification, but merely tries to pair off nodes with similar senses. Each successful match is assigned a score which depends on how recent the context node is and how close (in the WSN) the word senses of the paired nodes are to one another.[10] When the repetition criterion is applied to choose between competing MCS's, the weight given to an MCS is the sum of the weights of repetition matches involving pairings in that MCS. Thus if (6-34) occurred in a text after it had been stated that John hit someone, the repetition match of the "hit" node in (6-34) with that for the earlier "hit" action would pair "he" with John. On application of the repetition criterion, MCS1 would pick up some points from this match and MCS2 would not; the arbitrator would therefore prefer MCS1. More intuitively, we might say that SPAR reasons that if it has been told about John hitting someone, then in the absence of any other indications (which CSI might be expected to pick up) he is more than averagely likely, and therefore more likely than Bill, to do it again.

Repetition matching can also be seen as a rudimentary attempt to take account of the discourse topic of a story. In the remainder of text (6-33), the beginning of which we looked at above, repetition matching is the deciding factor in resolving the three underlined anaphors.

[1] John promised Bill that he would mend his car.
[2] He took it to his friend's garage.
[3] He tried to persuade his friend that he should lend him some tools.
[4] His friend said that <u>he</u> was not allowed to lend tools.
[5] John asked his friend to suggest someone from whom <u>he</u> could borrow tools.
[6] His friend did not answer.
[7] Fulfilling <u>his</u> promises was important to John.
[8] He was angry.
[9] He left.

In [4] and [5], the repetition criterion favours John rather than his friend because sentence [3] mentions tools being lent to John (the "him" in [3] being decided by CSI). In [7], John is favoured again because sentence [1] talked about John's promise. In all three cases, the pronoun could be resolved by a sufficiently powerful inference mechanism; however, the fact that the essentially linguistic repetition criterion is able to arrive at the same answers by quite different means is evidence for the shallow processing hypothesis.

If the repetition criterion does not select one MCS as superior, the second criterion applied is that of <u>focus retention</u>. The default action in any text would seem to be to maintain the focus: that is, to continue to talk about the same thing rather than to keep switching between foci. SPAR therefore calculates what the new discourse focus would be if each of the competing MCS's were approved. Those which would maintain the discourse focus are preferred. (No analogous preference seems to exist for maintaining the actor focus).

---

[10] The exact way in which this score is calculated is somewhat ad-hoc; however, the underlying principle seems sound.

In the absence of appropriate CSI (here, perhaps taking the form of script application) the focus retention criterion provides correct decisions in

(6-35) [1] John went to a restaurant.
[2] He asked the waiter for a curry.
[3] He ate it.
[4] He paid the cashier.
[5] He left.

For the underlined pronouns, Sidner's rules indicate an ambiguity between actor focus (John) and potential actor focus (the waiter and the cashier respectively) which may be resolved by CSI. However, if CSI fails, the pronouns can still be resolved by noting that only if they are resolved to "John" will the discourse focus (John) be maintained.

If the focus retention criterion is insufficient, a third criterion, the c-commanded pronoun (CCP) criterion, is applied. This criterion acts to prefer the MCS in which the greatest number of pronouns are c-commanded by a cospecifying noun phrase (either full or pronominal). Thus if none of the earlier, more powerful factors were effective, the CCP criterion would resolve the "his" in

(6-36) On his arrival, John realised that Bill was drunk.

to John rather than Bill, because the NP node for "John" c-commands that for "his", while that for "Bill" does not.

The reasoning behind the CCP criterion is based on the same Grice-derived principle that Reinhart used (see section 2.4) to reformulate the c-command constraint: "if a speaker has the means to express a certain idea clearly and directly, he would not arbitrarily choose a less clear way to express it". Thus when a pronoun still appears ambiguous after the application of every other knowledge source available, the hearer may assume that the speaker did not have any clearer way to express it: in other words, that the speaker did not have the option of using a non-pronoun. And the most obvious reason why the speaker would not have had that option is that had he used it, he would (under his intended interpretation) have broken the c-command constraint and thereby misled his hearer. Therefore it is likely that the pronoun in question is c-commanded by a cospecifying phrase of some kind.

Although this argument is similar in form to Reinhart's, it is quite independent of it in content. It relies only on the fact that the c-command constraint has been observed to be highly accurate and very strong; this fact is independent of Reinhart's explanation of it.

This motivated (although possibly incomplete) set of three collective criteria - repetition, focus retention and CCP - is usually capable of resolving pronouns which CSI leaves undetermined. However, they cannot always do so; and when they cannot, the weak distributive preferences suggested by Sidner for contextual candidates and further developed for intrasentential candidates in section 6.3 are applied. These preferences are better than nothing but the choices they make are not reliable. Alternatives to applying them irrevocably are discussed in chapter 10.

6.5 Summary and evaluation

In this chapter we have seen how SPAR's processing is directed. The processing framework is similar to Sidner's in as much as anaphor resolution is driven by focus-based rules that suggest candidate specifications. However, significant alterations and extensions have been made to Sidner's framework in order to achieve coordination in four areas: between the resolution of different types of ambiguity, between different knowledge sources, between intrasentential and contextual candidate specifications, and between the resolution processes of different anaphors in a sentence. The results of addressing these four problems are as follows; evidence for the conclusions reached is provided mainly by the stories processed, details of which are given in the appendices.

(1) The "principle of anaphoric success" is quite widely applicable in stories, and can be used to resolve many, though not all, word-sense and structural ambiguities as a by-product of anaphor resolution. Sentence readings differing structurally and/or in word senses are processed separately; which one is ultimately accepted is determined by an internal criterion, semantic density, and an external criterion, ease of anaphor resolution.

(2) Sidner's rules for definite anaphors in isolation are quite accurate and do not require major changes. "Inferred specification" can generally be detected without CSI, by allowing the information constraint to be broken in a controlled way. The extension of Sidner's PI rules to indefinite pronouns seems promising but has not been thoroughly tested.

Sidner's "normal mode" inference, which is capable of rejecting suggested candidates, should (given a Wilksian theory of semantics and CSI) use only linguistic knowledge, since constructive inference is unreliable even if only analytically true inference rules are used. When "special mode" inference, involving CSI, is required, it should be postponed as long as possible and then applied, if at all, to all anaphors at once.

(3) The best way to incorporate ISCs (intrasentential candidates) into Sidner's rules is not to write extra rules but temporarily to augment the focus register contents with ISCs. With a few exceptions, the focus preferences between ISCs and between ISCs and contextual candidates are very weak and should be applied only as a last resort.

(4) "Collective" factors should be invoked after "distributive" factors. Collective factors include syntactic constraints, CSI, and the criteria of repetition, focus retention and c-commanded pronouns. Of these, only syntactic constraints are able to overturn the conclusions of distributive factors, but in practice they seldom do so. The other collective factors are applied preferentially and are used to choose between maximal consistent subsets of the set of surviving predictions.

In a shallow processing framework, all is far from lost if CSI fails to return the results that a powerful CSI mechanism would, since the three remaining collective criteria often guide the system towards appropriate conclusions.

These conclusions provide support for the shallow processing hypothesis in at least three ways. Firstly, they suggest that complex inference is not often necessary for recognising "inferred specification" of full NPs. Secondly, it is not merely the case that Sidner's "normal mode" inference can be performed without the use of world knowledge, as shallow processing would require; in fact, world knowledge ***cannot*** , in a system using Wilksian semantics and inference, safely be used in normal mode inference. Thirdly, and perhaps most importantly, the thorough exploitation of purely linguistic collective factors often enables the same conclusions to be reached as would be reached by a powerful inference mechanism.

However, there are some ambiguities for which the methods presented here are inadequate. They include those non-anaphoric ambiguities not covered by the principle of anaphoric success; such ambiguities fall outside the scope of this work. (Some ways in which they might be dealt with are discussed in chapter 10). More serious are those anaphoric ambiguities which ultimately have to be resolved by applying unreliable weak distributive preferences; some possible solutions to this difficulty will be a main strand of chapter 10.

7. Inference and world knowledge

"He who increases knowledge increases sorrow."

Proverbs 1:18

The ability to makes inferences about the situations and events described in a text is an essential attribute of any program which is intended to understand natural language. The deeper the level of understanding required, the more sophisticated are the necessary inference processes.

Inference is necessary for intelligent processing of virtually any kind, and much research has been directed at it in all branches of AI; see, for example, Hobbs and Moore [1985]. However, given our concern with the specifically linguistic problem of anaphor resolution, we will restrict our attention to the use of inference in natural language understanding, and particularly in story processing. Emphasis will be placed on the way that inference brought about anaphor resolution. Further, as noted in chapter 1, the term "inference" (or, equivalently, CSI) will be used not in the very broad sense of deriving new information from old, but in the narrower one of finding the connections between propositions in a text.

We will see that the inference required for "open world" texts such as stories, where the possible types of object and event in the domain cannot even in principle be fully captured or formalised, is quite different from (and normally more complex than) that required for reasoning about closed, fully specifiable domains. Because this thesis is concerned with the processing, and in particular the anaphor resolution, necessary to paraphrase simple stories, the projects selected for discussion in this chapter are ones which have proved particularly influential in research into story understanding, and/or approach inference along the same general lines as SPAR, and/or tackle anaphor resolution in an interesting way.

I will not try to provide complete descriptions of the projects discussed, since most of them are well-known and all are fully described elsewhere; rather, I will concentrate on how they treated anaphora and how, if it all, their treatments of anaphora applied focussing criteria as well as inference. I will also attempt to demonstrate, given the immense complexity of full-scale, deep inference, the importance of considering an alternative, shallow processing approach for some tasks.

The many and varied knowledge structures and associated inference processes which have been proposed over the years, under names such as demons, scripts, plans, and MOPs, can be distinguished along at least four dimensions. These dimensions are in principle independent although in practice certain characteristics tend to appear together.

Firstly, inference may be <u>local</u> or <u>global</u> in range. Local inference consists of inferring a new pattern or proposition from a small number of old ones without reference to the story representation as a whole, while global inference takes a large part of the story representation into account and may lead to a global interpretation of all or part of the story (e.g. that it describes a restaurant visit).

Secondly, we may classify inference according to <u>content</u>, or the <u>class</u> of knowledge it uses: a particular system may specialise in applying knowledge about physical causation, about stereotyped situations, or about how people form plans.

Thirdly, the <u>mechanisms and structures</u> which have been used for inference are very varied. For example, a piece of knowledge may be encoded as a demon, a frame, or a discrimination net.

Fourthly, the knowledge exploited for inference may be more or less <u>abstract</u>; that is, it may be more or less directly related to what we expect to find explicitly stated in the text. This will affect the way it is mapped onto the text. The knowledge that "people often try to kill two birds with one stone" is more abstract than the knowledge that "visiting a restaurant normally involves opening the door, being shown to a table...".

Early story understanding work, such as Charniak's DSP and Rieger's MEMORY which we will examine first, tended to concentrate on local inference, largely about physical causation, at a low level of abstraction.

The inability of such systems to process stories of any significant complexity led to a series of research projects based, as MEMORY was, on Schank's Conceptual Dependency (CD) representation. Some of the resulting systems were <u>specialised</u>: they used a single mechanism to exploit a single class of knowledge, but did not assume that other classes could be exploited with the same mechanism. Inference was predominantly global, and increasingly abstract in successive projects. Other, mostly later, projects were <u>heterogeneous</u>: they used different specialised mechanisms to apply different classes of knowledge. Both global and local inference was performed; in some cases, knowledge was encoded at various levels of abstraction.

Although both specialised and heterogeneous systems performed quite impressively, the former were limited to texts of a particular type, while the latter were very complex. This prompted some researchers to develop homogeneous or <u>uniform</u> systems which represented and processed different classes of world knowledge, however abstract, in the same way. Among such systems, Cater's AD-HAC is described.

This description of the development of story understanding work in the CD paradigm provides a suitable background for the subsequent discussion and evaluation of Wilks' theory of inference, on which SPAR is based, and related work.

Quite other approaches to anaphor resolution have been proposed, notably by Mellish, and the chapter finishes with a description of his "incremental" approach to reference evaluation in mechanics problems. I argue that the type of inference appropriate for processing reference in such constrained texts is quite different from that required for "open world" texts.

Many of the projects chosen for discussion in this chapter are CD-based. This is partly because much of the important work in story understanding has been in the CD tradition, and partly for the sake of continuity in tracing the development of ideas.

In most of the systems discussed (Wilks' being the main exception), anaphor resolution is not treated as a distinct, separable part of the overall understanding process; rather, anaphors are resolved as a by-product of inference. All the systems covered assign a very minor role to local focus; typically, they use recency as an approximation to focus, and only apply even that when inference alone is inadequate. This can result in large amounts of very complex inference being performed. One of the main goals of this thesis is to show that in a system whose main purpose is anaphor resolution rather than, say, question answering or summarising, such complex inference is not often required for considerately-written texts.

7.1 Two early story understanders

Charniak [1972] and Rieger [1974,1975] were among the first to investigate the inference needed to understand stories.

Charniak

Charniak attempted to discover what inference was necessary to understand stories in sufficient depth to answer questions about them and to resolve anaphors. His DSP (Deep Semantic Processing) system accepted as input hand-coded *assertions* which together constituted the story representation. These assertions were simple predications such as might have been derived from sentence analysis. As assertions arrived, they were passed through four stages of processing. This processing might give rise to new assertions which were queued to be processed in turn.

The most important stage of processing consisted of applying demons. A demon was a piece of code which was dynamically created during the processing of an assertion matching a certain pattern. When created, it would wait for new assertions matching a second pattern. If and when such an assertion arrived, the demon would take some action, such as making a new assertion. For example, if it was asserted that it was raining, a demon would be set up to wait for an assertion that some person was outside. If it found such an assertion, it would assert that that person would get wet.

Anaphors were resolved during inference as follows. When an assertion was processed which contained a new token representing a pronoun or FDNP, a possible referent list (PRL) was constructed for that token. The PRL would consist of all possible referents mentioned earlier which were not ruled out on syntactic (configurational), semantic or gross recency grounds. If the newly-constructed PRL had only one member, it was accepted as the referent; if it had several, the anaphor would be represented by a variable which, when demons containing it were applied to assertions in the queue, could only be bound to a member of the PRL. After a certain time, any unresolved anaphor was taken to refer to the most recently-mentioned member of its PRL.

The defects of Charniak's model are well known, and motivated further research. We concentrate here on two which directly concerned anaphor resolution.

Firstly, when several demons could apply to an assertion to resolve an anaphor in it in different ways, the most recently created one would always resolve it. However, this heuristic will not in general lead to correct results.

A second drawback, which DSP shared with all early systems, was that focus was not properly treated. An unfocussed candidate was preferred over a focussed one whenever required by a demon. Charniak considered an alternative to this: a "backup method", in which candidates are considered one at a time and the first plausible candidate is accepted. He presented evidence that neither recency or topichood provides an adequate ordering for such a method. However, his argument assumed a rather narrow definition of plausibility and very straightforward algorithms for ordering candidate referents, and is therefore not convincing.[1]

Rieger

Rieger's MEMORY improved on Charniak's DSP in allowing interaction between uncertain inferences; in DSP, uncertain inferences (in the form of demons) did not interact. Furthermore, MEMORY's use of the Schank's primitive-based Conceptual Dependency (CD) representation allowed its inference procedures to apply to a wider range of patterns than Charniak's had done, giving it greater generality.

Rieger developed MEMORY as the inference component of the MARGIE story understanding system (Schank [1975]). MEMORY accepted CD representations of structures for single sentences from Riesbeck's English analyser and made inferences in order to establish the connectivity of a text. These inferences were extensively cross-referenced, making MEMORY's representation more integrated than DSP's list of assertions. They could be expressed in English by Goldman's sentence generator BABEL.

Processing in MEMORY consisted of a ***relaxation cycle*** in which reference resolution and inference were performed alternately. In the reference phase, recency and semantic constraints (both explicit and inferred) were used to eliminate candidate referents for an anaphor much as in DSP; however, a firm identification, resulting in an anaphor being merged into a candidate referent, was made only if exactly one candidate was clearly superior. Otherwise the decision was postponed. In the inference phase, all original and inferred conceptualisations judged sufficiently interesting (on the basis of their content and their "strength", i.e. the certainty with which they were believed) were used to trigger inference. Reference resolution would often allow two inferences to be recognised as identical in content, thus completing a causal chain; however, inference procedures were of sixteen distinct types, and only certain combinations of types could form chains.

Reference resolution sometimes enabled further inference because the merging of two concepts made some inferences valid for the first time. Inference in turn enabled reference resolution to advance because it produced further semantic constraints.

---

[1] Specifically, the argument assumed that the ordering used for a pronoun in a direct quotation is the same as that used for one in normal text. However, it is arguable that when processing a quotation, the reader tries to model not the writer's focus of attention but that of the character being quoted. A different candidate ordering may therefore be appropriate.

Thus whereas in DSP a single demon was allowed to resolve an anaphor, MEMORY's approach was more cautious. The effects of all relevant inferences were considered together during the reference phase, and the anaphor was only resolved if one candidate clearly stood out as superior to the others.

However, MEMORY's main defect was that its inference was too undirected. The effort to construct long causal chains meant that great numbers of irrelevant inferences were produced. This was largely because the knowledge encoded in inference molecules was all local and mainly causal; there was no global inference able to perceive larger patterns. Consequently, MEMORY was defeated by stories of more than a few sentences.

One possible way to inhibit the explosion of inferences, at least for some tasks, is to adopt Sidner's proposals and let focussing constrain inference. Another way (which could be combined with the first, but in the CD tradition at least, has not been) is to provide the system with knowledge in a more structured form. Later CD-based programs, which we will examine in the next two sections, did just this.

7.2 Specialised and heterogenous approaches to story understanding

Cullingford

Cullingford's [1978] SAM (Script Applier Mechanism) was a specialised system which exploited knowledge about stereotyped situations to understand stories describing such situations. It accepted input from Riesbeck's ELI analyser and was able to summarise it, translate it and answer questions about it. SAM prevented the combinatorial explosion of inference from which MEMORY had suffered by mapping scripts onto a text. A script encodes knowledge, in the form of one or more prepackaged causal chains, about the typical features of an activity or event such as a car accident or a cinema visit. In SAM, a script would be activated by the arrival of input matching certain of its patterns; subsequent input would then be matched against the patterns in the activated script.

A script also included a set of ***roles*** representing the people, objects and places participating in it. When a script was activated, some of its roles were bound to story participants; other roles were assigned default values, so that later references to them could be understood. Thus in

(7-1) John entered a restaurant. He asked the waiter for the menu.

both the underlined FDNPs would be understood as references to roles in the restaurant script. Pronouns were resolved as a by-product of matching events with patterns in the script; if the above story continued

(7-2) He ordered a hamburger.

the pronoun would be interpreted as referring to John and not the waiter because ordering food is specified in the script as an action of the customer - in this case, John.

SAM was able to handle longer and more realistic stories than MEMORY, including many taken from newspapers. However, although it was capable of some MEMORY-style low-level inference to bridge gaps between story events and script patterns, it was defeated by stories which could not be accounted for by single scripts (and, of course, by stories which required non-script-based inference). SAM raised, but did not solve, the difficult problems of how to decide when a script should be activated, when it is no longer relevant, and how script interactions should be managed. Tait [1982] attempts to solve some of these problems.

SAM did not make explicit use of the concept of focussing, although one might identify the global focus of a script-based story with the currently active script(s). In SPAR, local focus sometimes acts to disallow otherwise plausible inferences; although these inferences are not usually scriptal, it may be that a local focussing mechanism could provide useful guidance when deciding what script(s) were applicable at a given point in a story.

DeJong

SAM was very slow and quite fragile, partly because it tried to understand every detail of the stories it processed. In contrast, De Jong [1979] provided his FRUMP system with "sketchy scripts" which were much less detailed than SAM's, containing patterns only for the most important events in a stereotyped situation. FRUMP did not attempt to understand or even to parse all its input, but aimed only to recognise the essentials specified in sketchy scripts. The significant assumption that all the essential information in a text would match scriptal expectations was perhaps more justified for the routine newspaper reports processed by FRUMP than for other genres, in which the stereotyped events tend to be less interesting than the unusual, unexpected ones. FRUMP's ignoring all but the essentials of a text enabled it to provide quite accurate, if very basic, summaries of a wide range of real news reports. However, like SAM, it was limited to stories corresponding to its scripts; and anaphors were resolved, if at all, as a by-product of script application.

FRUMP's processing strategy has something in common with SPAR's. Both systems operate using incomplete knowledge. FRUMP's success showed that complete understanding is not essential to accurate performance of some tasks. However, FRUMP's processing was not so much shallow as partial; much of the input was ignored altogether, while the rest was processed at a fairly deep level, using quite specific world knowledge. In contrast, SPAR processes all its input, but mostly at a shallower level than that of script application. This important difference corresponds to the difference in the tasks the two systems perform. To generate summaries, FRUMP had to identify the important events and their role in the text as a whole, and could ignore everything else; SPAR generates paraphrases, and must therefore process everything in the input, although not necessarily in such depth.

Wilensky

One large class of stories for which scripts (as ordinarily formulated in terms of stereotyped event sequences) alone are inadequate is those involving the goals of actors and the often novel plans they formulate to achieve those goals. Wilensky's [1978] PAM (Plan Applier Mechanism) was aimed at understanding such stories. PAM resolved pronouns as a by-product of inference which related an actor's plans, goals and actions and traced the progress of a goal towards fulfilment or frustration. The stated or inferred goals of an actor gave rise to certain expectations about his actions, and, just as in SAM, the matching of a subsequent sentence representation with an expectation often entailed resolving a pronoun in a particular way.

PAM was a more flexible system than SAM because it did not require stories to follow a rigid pattern. However, this greater flexibility was accompanied by considerable complexity in the knowledge represented and the processes using it. Furthermore, like SAM, PAM was a specialised system, limited to understanding stories of a particular type (in PAM's case, those involving the goals of a single actor). Subsequent research in the CD paradigm identified other varieties of knowledge which were essential for full comprehension of some stories. Attempts were therefore made to build heterogeneous systems which could apply knowledge structures varying widely in both form and content to a story in a coordinated fashion.

The BORIS system

One such system was BORIS (Lehnert et al [1983]), an experimental program which used over fifteen different knowledge sources to understand, in depth, realistically complex stories of two to three hundred words. It was capable of answering a wide range of questions about the events in the stories and the reasons for them. The system's knowledge base consisted of interlinked structures of various kinds. These included MOPs (Memory Organisation Packets), which can be seen as generalisations (and in fact abstractions) of scripts and contain information about goals and intentions as well as events, and still more abstract TAUs (Thematic Affect Units) which represent knowledge about how generalised event patterns such as a "close call" or a "broken obligation" affect people's emotions. Activated knowledge structures could be combined and conflated to generate predictions about subsequent events. These predictions were used not only to integrate text sentences into the story representation but also to fill in missing semantic roles, resolve pronouns and disambiguate word senses.

BORIS showed clearly the immense complexity of the inference involved in in-depth understanding and the difficulty of specifying, encoding and using the necessary knowledge. Although the theory behind BORIS is in principle applicable to a wide range of stories, the implemented system as described in Lehnert et al [1983] was only capable of understanding two or three different stories.

The history of CD-based story understanding is, with the notable exception of FRUMP, one of successively more flexible and complex inference mechanisms using an increasing number of ever more abstract kinds of knowledge. In spite of the significance of BORIS and many of its predecessors in showing what deep inference might involve, it would appear that for the foreseeable future, "robustness" (in the sense of being able to cope with variations of stories already processed) and in-depth understanding are mutually exclusive in a language processor, at least for "open world" texts.[2] In response to this, the purpose of the research described in this thesis is to discover how far it is possible to avoid such complex inference in a system which performs a primarily language-based task such as paraphrase.

The complexity and limited applicability of many of MEMORY's successors led some researchers to question whether each type of knowledge used by a story understander had to be represented differently: that is, whether a wide difference between the contents of two knowledge structures had to imply a similarly wide difference between their forms and the ways they were used. Two systems adopting the alternative uniform approach will now be discussed.

7.3 Uniform approaches to story understanding

The AD-HAC system (Cater [1981]) was a complete CD-based story understander whose inference component represented an attempt to overcome the problems of Rieger's MEMORY while avoiding the complex, special-purpose and widely varying knowledge structures used by SAM, PAM and their successors. As in MEMORY, the inference component alternated between generating new inferences and identifying pairs of inferences ("compacting"); inference was controlled by interest ratings, and certainty (confidence) ratings were used to deal with incompatibilities. However, the construction of *long* causal chains did not play such a central role as in MEMORY, and AD-HAC jettisoned MEMORY's unrestricted inference procedures in favour of more structured inference networks, which were based not only on conceptual primitives but also (to allow the kind of inference for which SAM used scripts) on classes of object. Inference networks are generalisations of discrimination nets, and are described below. They provide a uniform representation for both causal and stereotyped knowledge, and potentially for other types (e.g. goals, plans and thematic affects) as well.

AD-HAC was capable of resolving the pronouns in, and answering questions about, stories several sentences in length. When the inferencer received conceptualisations from the analyser, it fed each one to the entry point of the inference network associated with its head primitive. As the conceptualisation passed through the network, certain ***actions*** were performed such as inferring new conceptualisations and marking certain entities as referentially distinct. The path taken through the network depended on the outcome of ***tests***, both on the semantic content of the conceptualisation itself, and on the wider story representation. If the result of a test was indeterminate (e.g. if neither a fact nor its negation was believed) both continuations were followed; the resulting alternative inferences were cross-referenced as incompatible with one another.

Processing by an inference network might be temporarily suspended by the issuing of a ***complaint***. Complaints were made when a role filler lacked a

---

[2] Of course, it might also be the case that for such texts, robustness is impossible *without* in-depth understanding. If so, robust processing of open world texts is altogether impossible for the foreseeable future. However, this is a conclusion we should not adopt without being forced to it.

necessary semantic feature; for example, the INGEST network demanded an unambiguously animate agent, so that the "they" in "They ate some bananas" would provoke a complaint. The network could only be restarted with the offending pronoun replaced by candidate referents having the desired feature; if there were several such candidates, they gave rise to mutually incompatible alternative continuations.

When all inference networks had terminated, the story representation was *compacted*. Similar inferences with certainty ratings of the same sign and preferably large magnitude were merged. As a side effect of this, pronoun referents were selected. Merging of inferences resulted in an increase in their certainty and interest ratings; the effect of these increases percolated to other inferences via cross-reference links. In particular, any inferences marked as incompatible with the merged ones were made less certain.

Thus the relationship between inference and reference resolution was different in AD-HAC and MEMORY. In MEMORY, the reference phase compared the entire occurrence sets of (the representations of) anaphors and candidates, and accepted a candidate only if it matched better overall than its competitors. In AD-HAC, anaphors were resolved either as dictated by the most certain pair of matching conceptualisations they took part in, or (less often) by the complaint mechanism excluding all candidates but one. The role of certainty and interest ratings in the whole understanding process was much greater in AD-HAC than in MEMORY.

AD-HAC was able to process longer texts than MEMORY was without the introduction of multiple knowledge structures. AD-HAC's inference networks with their complaint mechanism, and the sophisticated treatment of alternative sets of inferences by certainty ratings and cross-referencing, suggest that systems based on a single general inference mechanism need not suffer from the combinatorial explosions that afflicted MEMORY. However, from the point of view of anaphor resolution, AD-HAC's machinery was very heavy-handed, doing by inference work which could often be done much more easily by coordinated use of a focussing mechanism,[3] lexical knowledge and syntactic coreference restrictions, as this thesis attempts to show.

It might be objected that the advantages of a uniform approach are likely to be illusory, because the gain in simplicity in the inference mechanism will be offset by extra complexity in the knowledge base. However, Norvig [1983], whose FAUSTUS system is another uniform CD-based program, argues that while in FAUSTUS "the complexity has not disappeared; it has merely moved from the processor to the knowledge base", uniformity gives greater extensibility and flexibility. Whereas an extension to a heterogeneous system will typically only improve the way the system processes one kind of knowledge, an extension to a uniform system will have a more global effect.

The performance of AD-HAC, and indeed FAUSTUS, suggests that a system which applies many types of world knowledge need not use a similar number of

---

[3] Interestingly enough, Cater states that AD-HAC resolves one difficult pronoun in an example story not as a result of inference but because one candidate referent is "more in focus" (Cater [1981], p179). However, the focus mechanism is not described.

knowledge structures and processing mechanisms. However, the evidence is not conclusive because these systems have not (yet) been shown capable of processing stories as complex as those understood by e.g. BORIS.

7.4 Inference and preference semantics

It is natural to ask, given the complexity of "deep" understanding, whether an alternative shallow approach might be equally successful for some tasks. We will now therefore look at Wilks' relatively shallow preference semantics theory of inference. This theory has a number of weak points, which will become apparent below. However, the principles underlying it are attractive; and because it is the basis of SPAR's inference mechanism, it will be discussed in some detail, and then compared with the closely related theory of Hobbs [1976].

7.4.1 Wilks' theory of common sense inference

Wilks divided definite pronouns into three types, A, B and C, according to the type of processing that was required to resolve them.

Type A pronouns were ones which the PS system could resolve by special anaphora paraplates in the paraplate matching procedure described in section 3.2. In Wilks' [1975b] example

(7-3) Give the bananas to the monkeys although they are not ripe, because they are very hungry.

paraplate matching procedures inspected the formulas for the adjectives "ripe" and "hungry" and saw that they prefer to be applied to plantlike and animate entities respectively. These preferences were satisfied only by interpreting the first "they" to mean "the bananas" and the second to mean "the monkeys".

If the preference criteria involved in paraplate matching were insufficient to resolve all pronouns, PS entered <u>extended mode</u>. The process of <u>extraction</u>, as shown below, was applied to every template containing the formula of the pronoun or a possible antecedent. Extraction consisted of "the unpacking of every possible case tie: both those in the formulas of the template and those labelling a link to other templates" (Wilks [1975b]). It resulted in the construction of further templates which were in some sense analytically implied by existing ones. Pairs of templates, both old and new, were then matched together in the hope of binding the pronoun to a possible antecedent. Thus PS resolved the "it" in

(7-4) John drank the whisky from a glass, and <u>it</u> felt warm in his stomach.

as follows. Template and paraplate matching resulted in the single-bracketed templates (T1 to T4) in figure 7.1 being constructed, with case ties linking T1 to T2 and T3 to T4. The formula for "drink",

```
DRINK1:   ((*ANI SUBJ) (((FLOW STUFF) OBJE) ((SELF IN)
               (((WRAP THING) FROM) (((*ANI4 PART) TO) (MOVE CAUSE)))))
```

T1: [John drank the+whisky]

E11: [[the+whisky (IN in) John+PART]]

DIRE

E12: [[the+whisky (DIRE to) John+PART]]

T2: [∅ (DIRE from) a+glass]

E21: [[the+whisky (DIRE from) a+glass]]

T3: [it felt warm]

IN

T4: [∅ (IN in) his+stomach]

E41: [[it (IN in) his+stomach]]

Figure 7.1: PS representation for "John drank the whisky from a glass and it felt warm in his stomach"

specified that the stuff drunk ("(FLOW STUFF)") ends up in the drinker ("SELF"); the extraction E11 could therefore be made. (The IN in E11 indicates that it is the location sense of the word "in" that is being used here. Extracted templates are indicated by double square brackets). The extraction E12 arose in a similar way. The IN case tie linking templates T3 and T4 justified the extraction E41.

When extractions had been made, type B pronouns would yield to a simple "zero point" matching strategy in which pairs of (original and/or extracted) templates containing the pronoun and a possible antecedent respectively were matched by comparing the formulas in corresponding slots. For (7-4), extractions E11 and E41 matched because the formulas for "the whisky" and "it" matched, as did those for "John+PART" and "his stomach" (the action slots for the two extractions being identical). PS accepted this match, thereby deciding that "it" referred to the whisky. (The other consequence of accepting the match, that "his" referred to "John", was of little interest since it only confirmed what had presumably already been decided by paraplate matching).

However, in situations where no (useful) zero-point match could be made, Common Sense Inference Rules (CSIRs) were invoked. A CSIR consisted of a pair of template patterns (i.e. template-like objects, the formulas in whose

---

[4] Wilks has MAN rather than *ANI here, which is presumably an error since the entity in question must be the agent of the drinking.

slots could contain variables as well as primitives; see below for an example) which could be applied in one or both directions to generate a new (temporary) template[5] from an old one. Matches between templates were then sought as before. If no useful match occurred, another round of CSIR application could in principle (though in the implemented system, did not) take place, again followed by matching.

Wilks [1975b] described the way that PS used CSIRs to process the text

(7-5) John left the window and drank the wine on the table. It was good.

The following templates were derived directly from the text:

T5: [John left the+window]
T6: [John drank the+wine]
T7: [the+wine (LOCA on) the+table][6]
T8: [?it was good]

Extraction from T6 produced

E61: [[the+wine (IN in) John]].

CSIRs stored under the head element of the action formula (if any) of each of the above templates were then applied. One rule stored under CAUSE (the head primitive of the formula for "drink") was

(7-6) ((*ANI 1) ((SELF IN) (MOVE CAUSE)) (*REAL 2)) → (1 #*JUDG 2)

which can be paraphrased as "if an animate being causes a real object to move

---

[5]The sequence of inferences on Wilks [1975b,p69] shows the results of CSIR application in template form, but Wilks [1977b] says that "[CSI] rules do not produce fresh template-like forms the way extractions do". However, it would appear that the objects that CSIR application produces are template-like in their actor-action-object form, even if they differ from templates in being temporary and in containing objects such as variables, # signs and perhaps functions (e.g. " "*judges" [in the right hand side of a particular CSIR] is represented by a general function satisfied by actions of liking..." (Wilks [1977b,p242])). It would seem that in practice the templates in CSIRs usually contained formula-like structures, to be fuzzy matched with existing formulas in roughly the same way as formulas were matched during zero-point matching; but there was in principle nothing to stop them containing matching predicates of arbitrary complexity. However the concept of a CSIR is far from being a vacuous one because a CSIR, viewed as a procedure, was restricted to accessing the information in a single template at a time, and returning a single new template as a result.

[6]Although Wilks presents this template as an original one, it seems by comparison with the "whisky" example that it is in fact an extraction from a similar template with a dummy symbol in agent position. However this does not affect subsequent processing.

into him/her/itself, then that animate being may judge that real object".[7] When applied to T6, the resulting temporary template (marked by triple square brackets) was

CSI61: [[[John judges the+wine]]].

Inference from T8 proceeded as follows. First, a copy of T8 was created in which "?it" was replaced by the formula for "the+wine", as one of the candidate antecedents. (Presumably other copies representing the other possibilities were also created, but did not lead to successful chains). Next, the following rule stored under the primitive BE (the head of the formula for "is") was applied:

(7-7) (1 BE (GOOD KIND)) ↔ ((*ANI 2) WANT 1)

This gave the temporary template

CSI81: [[[*ANI[8] want the+wine]]]

which matched CSI61 to form a complete chain. The assumption made in creating the copy of T8 from which CSI81 was derived, that "it" referred to the wine, was vindicated by the fact that no chains of the same or shorter length were found, and the pronoun was resolved.

In the version of PS described in Wilks [1975b], no attempt was made to construct chains involving more than two CSIR applications. Wilks commented "This length limit could easily be extended, but I suspect that understanding of normal situations rarely requires chains longer than three" (p69). If CSIRs failed to resolve a pronoun, then "...the top level of the system tries to resolve the problem by default, or what a linguist would call focus. Roughly, that means: assume that whatever was being talked about is still being talked about" (p65). However, the working of this application of focus was not explained, if indeed there was any significant mechanism.

7.4.2 An evaluation of Wilks' theory

As argued in chapter 1, Wilks' theory of inference is attractive for two main reasons: firstly, its "partial" approach, according to which the inference mechanism is only asked to provide sufficient partial explanations to resolve specific ambiguities, rather than having to construct a global explanation of the events described by a text; and secondly, its simple heuristic (preferring shorter chains to longer) for deciding on the validity of a set of inferences.

But as Wilks admitted, the theory was not very extensively tested in its implementation in PS, especially as far as pronoun resolution was concerned. This was because the texts PS processed were either very short or presented no significant pronoun problems, and in any case the number of texts in either

---

[7] The "#" indicates that any NOT prefixes in primitives should be ignored during matching. The numbers 1 and 2 in the CSIR are variables which show how formulas in the source template should be used in the new one.

[8] Wilks has "John" instead of *ANI, but the information that the (*ANI 2) in the CSIR is in this case John is not available before a comparison is made with CSI61.

category presented in the literature was fairly small. The inference mechanism had four further drawbacks; however, these do not, in my view, outweigh the two fundamental advantages of Wilks' theory mentioned above.

(1) Wilks [1975b,p69] states that "much of the effort of the program is in the inexact matching of the template forms to the [CSI] rules", but he provides no significant description of what this matching involves. Similar remarks apply to the processes of extraction and of paraplate matching, which can involve, respectively, primitive-specific and paraplate-specific procedures as distinct from any general "extractor" or "paraplate matcher" functions. The content of these specific procedures is not stated to be subject to any constraints, making it difficult to construct texts which PS could not *in principle* interpret correctly. The problem of unconstrained or unspecified processing is especially serious in the area of CSI, because the implemented PS system contained very few CSIRs, and so it is fair to assume that it constructed very few distinct inference chains. It is thus not clear that the matching procedures used for CSI would have been extensible had a large number of rules and possible inference chains existed.

(2) The lack of a non-trivial focussing mechanism was also a potential drawback. Because PS only processed texts which were either very short or presented no significant anaphor resolution problems, it was able to assume that every entity in the semantic representation was sufficiently focussed to be a possible antecedent. However, for longer texts, this would often have led to wrong answers; the inference component would frequently be misled by a spurious chain of inferences into recommending a candidate antecedent which a human reader, or a system maintaining an adequate representation of focus or context, would never consider; and in PS (unlike SPAR), the results of inference, if unambiguous, were always accepted. Focus was only used to select between candidates when inference was unable to do so, and therefore could never override inference. In addition, the system would risk doing a lot of unnecessary work making inferences about insufficiently focussed possible antecedents, even if such inference did not lead to wrong answers. We may conclude that for longer texts, PS would have needed to give a more prominent role to focus, both to delimit candidate sets and to take part more actively in selecting correct antecedents.

(3) As Wilks recognised, "...it remains to be shown that a large body of common sense inference rules can be controlled, and that solutions can be found to pronoun problems without the chains becoming inordinately long. There may also be something inherently implausible about <u>always</u> preferring a rule chain of length, say, 3 to one of length 4. (Though preferring length 1 to length 4 might be clearer and more plausible). This difficulty may mean that preferences in the system will have to be weighted in a way they are not at present." (corrected[9] from Wilks [1976,p169]). This issue is addressed in the next chapter; see also Hobbs' use of "rule salience" below.

(4) Finally, PS's inference mechanism apparently did not take account of the possible falsehood of statements in intensional contexts. The examples Wilks gives of CSIR application do not involve intensional contexts, and in fact

---

[9] In the original, the numbers 3 and 4 are the other way round, as are the numbers 1 and 4; but the meaning seems clear.

ignoring intensional context in short texts of the type Wilks processed does not often lead to a Wilksian inference mechanism misresolving pronouns. Quite often, such a mechanism will get a plausible answer (choice of antecedent) from a wrong chain of reasoning. For example, when processing the sentence (from Wilks [1975a])

(7-8) The soldiers fired at the women, and we saw several of them fall,

PS decides that "them" refers to the women, by using a CSI rule stating that entities who are shot at are likely to fall. There is no reason to suppose that it would not make the same decision, for exactly the same reasons, if the first clause read "The officer ordered the soldiers to fire at the women". Although the "them=women" decision is still plausible, PS would fail to realise that it was not asserted that the soldiers actually obeyed the order, and that assuming that they did should really be a common sense inference in its own right. The reason that plausible decisions are still made in spite of the system failing to distinguish what is asserted as true from what has to be assumed, is that guessing that an event in an intensional context gets realised is usually quite reasonable. That is, people's beliefs are often correct, their statements are often true, and they often do what they want to do. However, an inference mechanism that chooses the right answers for the wrong reasons is not theoretically satisfactory, and there is no reason to suppose that it would continue to deliver right answers for more complex texts.

Thus in summary, Wilks' preference semantics provides a very promising, though inadequately tested, approach to anaphor resolution. Its advantages are its shallow (and therefore not over-demanding) approach to semantic representation, its policy of invoking common sense inference only when necessary for specific ambiguity problems, and its prefential chain-length heuristic for choosing between inferences. Its drawbacks for anaphor resolution, which however are not so serious as to warrant abandoning it, are its unconstrained and/or undocumented use of data-specific pattern matching procedures, its lack of an adequate focus mechanism, its insensitivity to intensional contexts, and possibly the untestedness of the chain-length heuristic.

One question we have not addressed here is whether Wilks' basically local and causal inference mechanism can be extended to other types of inference, such as that performed by the systems described in sections 7.2 and 7.3. In the next chapter we will see how SPAR implements the beginnings of such an extension.

7.4.3 Hobbs' theory of inference

Hobbs [1976] suggested not only the syntactic pronoun resolution algorithm described in section 5.3.2 but also a "semantic" (i.e. CSI-based) theory which used preference-like principles and was relatively shallow. It is not clear how extensively, if at all, the theory was implemented and tested. Hobbs, like Wilks, invoked Joos' "semantic axiom number one", which he stated as

> The important facts in the text will be repeated, explicitly or implicitly.

In Hobbs' theory, the inference mechanism would accept hand-coded predicate calculus-like input such as might have been derived from a text by syntactic and semantic interpretation rules. With each predicate was associated a collection of inference rules. These rules were divided into clusters roughly according to their topic; the use of a rule served to increase the salience of all rules in its cluster. The mechanism attempted to discover the connectivity of the text by applying rules and merging the resulting conclusions where redundancy was recognised. Like PS, Hobbs' mechanism favoured chains involving fewer rules, but unlike PS, it also favoured those involving salient rules, thus making inference context-dependent.

The four primary semantic operations were:

(1) Detecting or verifying the intersentence connectives (i.e. coherence relations), approximately as discussed in 5.2.2.

(2) Predicate interpretation. A statement of the form "X walks out" would be encoded as `out(walk(X))`. The lexicon entry for "out" stated that an inference of the form `go(z1,z2,z3)` must be drawn (i.e. proved) from its argument. Any intermediate stages in the proof were instantiated (asserted). This strategy resulted in a great reduction of the number of senses stored for each word, and also allowed omitted material to be recovered, as in "(The price of) meat is high this month".

(3) Knitting. Whenever the predicate of a newly instantiated statement was identical to that of an existing one, the mechanism assumed a redundancy, and merged the two (and their corresponding arguments) unless an obvious inconsistency was found. This might result in pronouns being resolved.

(4) Identifying entities. If any definite pronoun remained unresolved after (3), the mechanism initiated a bidirectional inference search from statements involving the pronoun and statements involving the antecedent predicted by Hobbs' syntactic algorithm (see 5.3.2). This search might result in "proofs" of different antecedents; if so, the simplest proof was accepted. Hobbs does not say what happened if no proofs were found.

Hobbs' inference mechanism has many similarities to Wilks'. "Predicate interpretation" is equivalent to extraction since it involves drawing out the (usually analytically true) implications of the use of particular words. Operation (4), "identifying entities", consists of applying (apparently non-analytic) rules in salient clusters to relevant statements, rather like Wilks' CSIR process.

Hobbs' mechanism is perhaps an advance on Wilks' in its use of cluster salience and in using the syntactic algorithm's suggestions to constrain the bidirectional search, much as Sidner recommends that focussing should constrain inference.

However Wilks' theory (as reported) scores over Hobbs' in having a more explicitly defined semantic representation. Whereas Wilks' use of primitives and formulas enables inexact matching between assertions in the representation, Hobbs does not specify how the corresponding relationships between his predicates could be established. Wilks' theory of inference must also be regarded as better tested than Hobbs', since, even assuming that the latter was properly implemented, the input it accepted was hand-coded rather than, as in PS, derived by automatic analysis of an input text.

7.5 Anaphor resolution in a fully-specified domain

The inference mechanisms discussed so far have all been intended to operate on "open world", in some ways unrestricted texts such as stories. However Mellish [1982] tackled anaphor resolution for the well-specified, "closed" domain of mechanics problems, and was therefore able to resolve anaphors using rather different inference processes.

Mellish developed his ideas in the context of the MECHO system (Bundy et al [1979]) which solved mechanics problems stated initially in English. In this restricted domain, it was safe to assume that the relevant information could be fully and exactly represented. Anaphors could therefore largely be resolved by means of absolute constraints rather than uncertain inferences.

Mellish aimed to resolve anaphors, or at least to reduce the size of their candidate sets, as soon as possible in the understanding process. Early anaphor resolution can, he claimed, provide useful information for other disambiguation procedures and help diminish unnecessary computation.

As constraints (predicates derived from parsing) accumulated during processing of a sentence, they were handed to an inference mechanism which performed "incremental evaluation" to resolve anaphors when possible. In Mellish's "extensional approach", an anaphor was represented by the set of candidate referents which could be shown to satisfy all the constraints associated with it. New constraints caused candidates to be eliminated. When only one candidate remained, it would be accepted; if the constraints eliminated all the candidates, it was assumed that a wrong path had been taken in parsing, and the system backtracked. Candidate elimination for one anaphor could propagate to another via a constraint mentioning them both; thus all anaphors were resolved in parallel.

Mellish's approach to anaphor resolution relies on a number of properties of the mechanics problems domain which are unlikely to hold for more general, "real world" texts. In such texts, one cannot assume

(1) that the lack of any constraint-satisfying candidate referents for a definite noun phrase indicates a wrong parsing decision. Indeed, it is doubtful that any absolute criterion could sensibly control parsing; rather, a preferential criterion that chooses between alternative parses, such as the principle of anaphoric success discussed in chapter 6, seems to be needed.

(2) that the existence in the text representation of more than one candidate referent satisfying the available syntactic and semantic constraints is grounds for postponing a reference decision; although this may be true for mechanics problems, which are generally short and stated as clearly as possible, it is not true in general, and a focus mechanism will (as Mellish acknowledges) be required to choose

between candidates.

(3) that enough absolute (as opposed to preferential) constraints exist to enable the resolution of one anaphor to have much effect on others.

Mellish's approach therefore does not seem to be applicable to story understanding. Since the points (1)-(3) above would probably apply to any attempt to resolve anaphors by strictly deductive means, it seems that the inference methods appropriate for constrained, "closed world" texts cannot in general be extended to "open world" ones. The idea of constraint propagation, though, is still useful for story processing: as we saw in 6.4.1, SPAR sometimes uses configurational constraints to propagate the resolution of one pronoun to the resolution of another. However, firm constraints, whether syntactic or semantic, cannot be expected to be available to resolve every ambiguity.

7.6 Summary

The conclusions drawn from our discussion of inference so far are:

(1) Many classes of world knowledge are needed for story understanding: knowledge with various degrees of abstractness, about physical causation, plans, affects and many other areas.

(2) In-depth understanding, using all these classes, is too complex to be compatible with practical performance for the foreseeable future.

(3) However, this complexity may perhaps be reduced by using the same knowledge structure and inference mechanism for each class of knowledge: that is, to keep the same form regardless of content.

(4) Wilks' theory of inference suggests that the complexity may be further reduced if what is required is ambiguity resolution rather than full understanding.

(5) In any case, inference methods appropriate for processing texts about "closed" domains cannot easily be applied to story understanding.

A further claim, which the next chapter will attempt to substantiate, is:

(6) If anaphor resolution, rather than complete understanding, is desired, the complexity and quantity of the necessary inference and world knowledge can be greatly reduced without serious loss of accuracy, by making thorough use of linguistic knowledge.

8. Inference in SPAR

> "I have never yet been able to perceive how anything can be known for truth by consecutive reasoning - and yet it must be."
>
> *John Keats, letter to Benjamin Bailey, 1817*

This chapter is about SPAR's inference mechanism.

Section 8.1 examines some of the issues that arise from the shallow processing requirement (see 1.2) that SPAR have access to only limited amounts of world knowledge. This requirement turns out to have implications both for the way in which world knowledge should be encoded, and for the mechanism that is appropriate for exploiting that knowledge. It is argued that Wilks' model of inference is a promising basis for SPAR's inference component.

The rest of the chapter describes and evaluates this inference component. The evaluation is designed to answer the following questions:

(1) Do Wilks' basic ideas on inference, which were not thoroughly tested in PS, seem plausible under closer examination and more thorough testing?

(2) Does Sidner's framework significantly reduce the difficulty and amount of CSI needed for anaphor resolution, as she claims? If so, how? In what ways can Wilksian inference be further controlled, independently of focussing, so as to maintain accuracy and efficiency?

(3) How can a Wilksian inference mechanism be extended to make best use of the limited amounts of general world knowledge that are available to it in a shallow processing system? How can it take maximum advantage of the available linguistic knowledge, especially that provided by word sense formulas? In terms of the classification of inference mechanisms given in chapter 7, does extending the mechanism in these two ways make it specialised, heterogeneous or uniform?

Because SPAR is built to test the shallow processing hypothesis, its inference mechanism is not intended to be a powerful, general purpose inference engine; so I will not attempt to evaluate it from this point of view. It should also be stressed that I am not attempting to address the global problem of how inference should be done in story understanding, but only the more specific one of how a Wilks-based inference mechanism in a Sidnerian, shallow-processing system can help to resolve anaphors in stories. This is nevertheless an interesting problem because a full or partial solution to it would provide support for the shallow processing hypothesis and for both Sidner's and Wilks' theories.

Sections 8.2, 8.3 and 8.4 describe the main features of SPAR's inference mechanism, and in so doing attempt to provide some answers to questions (1), (2) and (3) respectively. Thus section 8.2 sets out the way that the basic

Wilksian procedure of constructing and completing inference chains is implemented. Section 8.3 concerns the way in which the inference process is controlled, firstly by means of a development of Wilks' previously unimplemented ideas on classifying inference rules, and secondly by using the partial results of the focus-based AR (anaphor resolution) rules. In both these sections, Wilks' ideas are seen to work quite well and are accepted without major modifications. However, some development of them is needed to allow some types of inference which Wilks' implemented system did not perform; section 8.4 therefore describes how the mechanism has been extended to incorporate inference about stereotyped situations, inference about actors' goals, and very weak, very general inference rules which are useful as a backup to the normal mechanism. The points made in the chapter are summarised in section 8.5.

8.1 Shallow processing, world knowledge and inference

The main aim of the work reported in this thesis is to test the version of the shallow processing hypothesis stated in chapter 1 as follows.

> A story processing system which exploits linguistic knowledge particularly knowledge about focussing, as heavily as possible, and has access only to limited quantities of world knowledge, which it invokes only when absolutely necessary, can usually choose an appropriate antecedent for an anaphor even in cases where the inference mechanism by itself cannot do so.

In order to motivate the design of SPAR's inference mechanism, we need to characterise more exactly the nature and extent of the "limited quantities of world knowledge" with which the system may legitimately be provided, and to outline the implications this characterisation has for the design of the inferencer.

The major reason why full, unrestricted CSI is still beyond the state of the art is that it requires vast amounts of complex world knowledge. In contrast, the amount of linguistic knowledge required for a language processor to give reasonably accurate performance on a task that does not require significant world knowledge is far smaller, and many quite reliable techniques exist for exploiting linguistic knowledge. Some of those techniques are adopted in SPAR. It therefore seems plausible that as long as the amount and complexity of the available world knowledge do not greatly exceed those of the available linguistic knowledge, then it should be possible to apply that world knowledge robustly and reliably whenever the available knowledge is in principle sufficient to solve the problem at hand. Of course, in cases which require knowledge that the system does not have, the inference mechanism cannot be expected to reach the right conclusions, no matter how robust and reliable it is as a mechanism; in such cases, it is up to the other, linguistic components of the system to try to compensate for the lack and guide the system towards the right decisions. Such compensation should normally be successful if, as the shallow processing hypothesis assumes, texts tend to be constructed considerately, so that constraints on interpretation derived from different knowledge sources confirm rather than conflict with one another.

Given the approximate constraint that the system's world knowledge should not greatly exceed its linguistic knowledge in amount and complexity, what guidelines should we follow in deciding exactly what world knowledge the system is to have? Clearly, we should try to make every piece of knowledge "pay its way" in terms of usefulness as far as possible. This means that a piece of knowledge (e.g. an inference rule) (1) should be as widely applicable as possible, and (2) should be represented as simply, and therefore as compactly, as possible. More specifically:

> (1) A given piece of knowledge should be as ***general*** as possible; that is, it should be relevant to a wide range of situations. This principle has at least three applications. Firstly, to the extent that it is possible to decide what concepts are more common than others in non-specialised texts, we should aim to provide more knowledge about common concepts than about uncommon ones. Secondly, when deciding what knowledge to provide about a given concept, we should not aim for completeness but should ask ourselves what information is most often likely to be useful. Thirdly, we should not provide specific information which could be inferred from more general. For example, if the system knows that people often pay for things they are given, it may be able to resolve the ambiguities in a story about a restaurant visit without being told specifically that paying is a normal part of visiting a restaurant.
>
> (2) Knowledge should be encoded as ***simply and succinctly*** as is consistent with reasonably reliable application. This has two implications. Firstly, we should not aim to specify completely the conditions under which a piece of knowledge is applicable. For example, we might tell the system that if someone makes a promise, he can be expected to try to fulfil it. Clearly, there are situations where this inference is unlikely to be valid: if the person is untrustworthy, or if something more urgent comes up. However, as long as such situations do not make the exception more likely than the rule, we should not attempt to encode them. Secondly, we should not aim for completeness in specifying causal relationships. The causal link between making a promise and trying to fulfil it is arguably mediated by goals such as those of adhering to moral principles or maintaining one's reputation; however, if knowledge of these intermediate stages is not normally required to resolve ambiguities in stories about promising, then it should be left out.

The need for world knowledge to be represented simply and succinctly, and the fact that SPAR is intended to process non-specialised texts, have important implications for the kind of inference mechanism that is appropriate.

Our discussion of the progress of research into story understanding led us to conclude that a ***uniform*** approach to inference - one in which many varieties of world knowledge are represented and in the same way - is the most promising one for our purposes. A ***specialised*** approach, in which only a few types of world knowledge are considered, is unlikely to be adequate; and a ***heterogeneous*** approach, such as that adopted in BORIS, is likely to prove too complex to perform robustly on stories on a variety of topics.

The (single) processing mechanism used by a uniform system would, in order to achieve generality over types of world knowledge, have to be fairly simple.

This requirement dovetails well with our requirement (2) above, that world knowledge should be represented simply and succinctly.

However, if, in accordance with (2), only the bare essentials of each piece of world knowledge are represented, then wrong inferences will quite often be made. The inference mechanism must therefore be able to hypothesise alternative lines of reasoning, rather than merely to follow a single one in which it has full confidence; and it must be able to choose between alternatives when they are inconsistent.

For these reasons, the model of inference embodied in Wilks' PS system seems a very promising basis for our inference component. Wilks' CSI rules were, in PS, represented simply as pairs of template patterns; and discrimination between alternative inference chains on the basis of chain length was central to the theory. However, the world knowledge used by PS was mostly straightforwardly causal and not very abstract; it has therefore been necessary, in SPAR, to incorporate a wider range of knowledge types, for example a certain amount of knowledge about stereotyped situations, without making the mechanism too complex or heterogeneous.

8.2 The basic CSI mechanism

In this section I will set out the way Wilks' basic mechanism of applying inference rules to build chains is implemented in SPAR and adapted to SPAR's representation which, as we have seen, differs in many ways from Wilks'. Having seen how and when extractions (analytic inferences) and common sense inferences are made, we will look at the circumstances under which a match between two inferences (either confirmatory or contradictory) warrants the formation of a chain. At several points, I will suggest solutions to some important problems not discussed by Wilks. Implementing these solutions does not involve making major changes to Wilks' framework; it will therefore become apparent that broadly speaking, Wilks' mechanism stands up well to closer examination.

8.2.1 Extraction and CSI

Just as in Wilks' PS system, the process of building inference chains in SPAR consists of two basic operations. The first operation is that of recursively making inferences from, on the one hand, story assertions involving undetermined anaphors, and, on the other, story assertions involving candidate antecedents. The second operation is that of trying to identify pairs of inferences so as to form chains which link assertions of the first type to those of the second and which thereby bind one or more anaphors to candidate antecedents. We will see later why it is necessary to infer both "backwards" from assertions involving anaphors and "forwards" from assertions involving candidates and combine the results to form chains, rather than just, say, inferring backwards from assertions involving anaphors and matching the results directly with assertions involving candidates.

All the matching of rules to inferences and of inferences to one another is done using the structure matcher described in chapter 4.

The following terminology, some of which has already been introduced, will be used to describe the inference process in both SPAR and PS. The assertions from which inference begins will be called starting points; those mentioning an anaphor are anaphor starting points, while those mentioning a candidate are candidate starting points. An inference is a proposition inferred, using an inference rule, from a starting point or from an inference already made. There are two types of inference rule: analytic extraction rules and non-analytic common sense inference (CSI) rules. The inferences made from applying these rules are extractions and common sense inferences respectively. A sequence of inferences originating from a starting point is a tendril (either an anaphor tendril or a candidate tendril depending on the starting point); a chain joining an anaphor starting point to a candidate one is constructed by successfully matching the inference at the end of an anaphor tendril with that at the end of a candidate tendril. (However, we will see that SPAR sometimes constructs a chain from an anaphor tendril alone.) Completing a chain involves making one or more bindings of anaphors to candidates.

In Wilks' PS, applying either an extraction rule or a CSI rule gave rise to a new template. The output of an extraction rule followed analytically from its input, so the templates output by extraction rules were as certain as their inputs and therefore could safely be made permanent. CSI rules, on the other hand, produced uncertain and therefore retractable conclusions, probably (see footnote 5 in 7.4.1) represented as temporary templates. PS tried first to construct "zero point" chains formed using only extraction rules, and if that was insufficient, searched for chains involving one CSI rule application, then two, and so on up to a preset maximum.

However, Wilks appears not to have considered the need for extraction rules to be applied to the outputs of CSI rules as well as to original templates. We will see in this chapter that it is frequently necessary to do so. The process of growing tendrils therefore should, and in SPAR does, consist not of applying extraction rules and then CSI rules but of applying extraction rules and CSI rules alternately. An inference made with an extraction rule does not increase the "length" of a chain for the purposes of the Wilksian heuristic that shorter chains should be preferred to longer; the "length" of a chain is a measure of its uncertainty, and extraction rules do not involve uncertainty. Chain length is therefore defined solely as the number of CSI rule applications involved.

If extraction rules are applied to the uncertain conclusions of CSI rules then the conclusions they produce, though by analytic means, are also uncertain. In SPAR, the inferences output by both extraction rules and CSI rules are therefore temporary, since they may be wrong; they are analogous in form to pieces of TMN, just as Wilks' CSI rules apparently output template-like forms analogous to the templates in the story representation proper. When a sentence has been processed and all its anaphors finally resolved, any inferences made by CSI are thrown away. Inferences only help to resolve anaphors, and are not used to augment the TMN permanently.

In PS, extraction rules were represented as individual procedures while CSI rules were pairs of template patterns which could typically be applied in either direction. Similarly, SPAR represents extraction rules procedurally and CSI rules declaratively, merely because the former tend to be more idiosyncratic in content and fewer in number. A CSI rule in SPAR (exemplified below) consists of a single pattern of TMN elements, with at least one pair of entry and exit points. If the input to a rule matches an entry point, the output will be constructed around the corresponding exit point. Most CSI rules can be applied in either direction, so that an entry point in one direction can be an exit point in the other. However, extraction rules are unidirectional, and this is reflected in their being implemented as procedures.

The basic inference process may be illustrated with two examples. The first involves only CSI rules; the second involves only extraction rules. Later we will see examples involving both.

<u>CSI example</u>.

When SPAR processes the text

(8-1) [1] The soldiers fired at the women.
[2] We saw several of them fall.

inference is needed to decide whether "them" refers to the soldiers or the women. The chain which leads to the decision that it was the women who fell consists of the following inferences, each joined to the last by a CSI rule. The English descriptions of each step (slightly modified from those produced by SPAR's generator - see appendix C) are as follows:

```
(1) SOME SOLDIERS FIRED AT SOME WOMEN.
              ↓
(2) A THING STRUCK SOME WOMEN.
              ↓
(3) SOME WOMEN WERE NOT ALRIGHT.
        matches:
(4) ref(THEM1-1) [1] WERE NOT ALRIGHT.
              ↑
(5) ref(THEM1-1) FELL.
```

A full listing of the CSI rules SPAR uses can be found in appendix D. The CSI rule by which (3) is generated from (2) is:

```
(STRIKHURT                       ; Name of rule (for debugging purposes only).
   ((RESULT S H <>))             ; The result of S may be H, where...
   (S STRIK (SUBJ P) (OBJE Q))   ; S represents P striking Q.
   (P *PHYSOB)                   ; P is a physical object
   (Q *PHYSOB)                   ; and so is Q.
   (H BE (SUBJ Q) (STATE WHOLE) (NEG NEG)))
                                 ; H represents Q being
                                   not whole (i.e. not alright).
```

The field (RESULT S H <>) relates entry point S to exit point H by saying that an event matching S may RESULT in one matching H. The <> indicates that the rule is bidirectional: i.e. that we may also use H as an entry point and S as an exit, to infer that something being "not whole" may be caused by its being struck.

[1] THEM1-1 is the TMN node representing "them". SPAR generates the pseudo-english phrase ref(THEM1-1) to avoid ambiguity.

However, not all rules work both ways; for example, while it is reasonable to infer if A has promised to do X that A will then try to do X, it is not reasonable to infer, even as a CSI, that if A is trying to do X then it is because A has promised to to X. This means that a strategy involving inference in one direction only (i.e. only anaphor tendrils, or only candidate tendrils) cannot be adequate. Since some inferences can only be made forwards (roughly, inferring effects from causes - see 8.3 below for more detail), and some only backwards, a mixed strategy involving growing both anaphor and candidate tendrils at the same rate is essential if all possible chains are to be found.

Extraction example.

Our second example is one requiring only extractions. For the text

(8-2) [1] John went to the zoo with Mary.
[2] They gave the monkeys some bananas, which they ate.

inference is invoked to decide whether the underlined "they" is the monkeys or John and Mary. SPAR assumes that the eating occurred after the giving (since, as indicated in chapter 1 when defining the task, all the stories processed describe events in chronological order). This time the steps in the completed chain are

(1) ref(THEY1-1) GAVE SOME BANANAS TO SOME MONKEYS.
↓
(2) SOME MONKEYS HAD SOME BANANAS.
matches:
(3) ref(THEY1-2) HAD SOME BANANAS.
↑
(4) ref(THEY1-2) ATE SOME BANANAS.

Extraction rules derive (2) as a necessary result of (1), and (3) as a necessary precondition of (4); (2) and (3) are then matched to complete a chain. The (procedural) rule used to derive (2) from (1) can be expressed in English roughly as follows: "If someone gives a physical object P to an entity E, then as a result, E has P".

Note that although the extractions themselves are analytic, the assumption that (2) and (3) describe the same state of affairs is not. (It is conceivable that the monkeys gave the bananas back, and then John and Mary ate the bananas). Thus, as argued initially in 6.2.3, even inference chains constructed from entirely analytic tendrils are not completely reliable, because an uncertain assumption is involved in the final match.

8.2.2 Completing chains

Clearly, therefore, we need a criterion for deciding when the two inferences matched to form a potential chain have enough in common for the assumption involved in the final match to be a safe one; mere consistency is not enough. For text (8-1), for example, the similarity between "ref(THEM1-1) were not alright" and "some women were not alright" was clearly enough to justify a chain being formed. But if we alter the text to read

(8-3) [1] The soldiers fired at the women.
[2] They were in the street.

then SPAR will match the inference "the women were not alright" with the representation of [2], and find that the two propositions are consistent. To form a chain on the basis of this match would be to make an unreasonable assumption, because the two propositions are only distantly related: one describes someone's location, and the other describes someone's state. Since people have locations and states all the time, there is no good reason to form a chain from such a match.

The heuristic SPAR uses for deciding whether two inferences are similar enough to justify forming a chain is that at least two of the bindings resulting from matching them should be <u>non-vacuous</u>. A binding is <u>vacuous</u> if the most specific WSN element dominating both nodes is a very common primitive such as BE, MOVE, or HAVE. That is, since most entities take part most of the time in states and events describable by such primitives, a match between two such states and events is wholly unremarkable.

When applied to the matches between the be inferences in our examples (8-1) and (8-3), this heuristic gives us the right results:

> (8-1): the two "be-not-alright" nodes are bound, as are the nodes for "them" and "the women". Both these bindings are non-vacuous, so form a chain.
>
> (8-3): the two "be" nodes are bound, as are those for "them" and "the women". But the first of these is vacuous (the two nodes' most specific shared ancestor is BE), so do not form a chain.

8.2.3 Single-tendril chains

There are, as mentioned earlier, circumstances in which a chain can be formed directly from a single tendril without the need for any match with a second one. This occurs when an inference (or a story assertion) is of a form which would be far more plausible if certain elements in it cospecified than if they did not: either because if those elements cospecified, then the inference would be almost tautologous (such inferences often instantiate "themes" in the sense of Schank and Abelson [1977]), or because if they did not, the inference would be unlikely to be true. Thus

(8-4) John promised Bill that <u>he</u> would mend <u>his</u> car.

contains two pronouns which SPAR resolves in this way. The first is handled by a chain which can be described informally as follows:

> (1) People usually make promises about their own actions rather than other people's, so "he" is probably John.[2]

---

[2]Note that this is only true of deliberate actions, not of events in general. The inference rule in question is not applicable to

(8-5) John promised his son that he would receive a computer for Christmas.

because receiving is not a deliberate action. Receiving the bicycle is not something that "he" will actively bring about; rather, "he" will passively have it happen to him. Note that if "receive" is altered to "buy", the inference rule does become applicable, and "he" is correctly interpreted as "John".

The second pronoun is resolved by a chain constructed along the following lines:

> (2) The use of (this sense of) "promise" implies that what is promised is something the recipient of the promise is likely to want; therefore Bill wants "he" to mend "his car"; therefore Bill wants "his car" to function. Since it is almost always true that people want their own possessions to function, but not necessarily other people's possessions, "his" is probably Bill.

Inference like this is performed by SPAR using CSI rules which have entry points without corresponding exit points, which are unnecessary. These NO-EXIT CSI rules are applied like any others, by matching the input with the entry point. For a NO-EXIT rule, however, a successful match with an entry point leads not to a new inference but to a completed chain. The inference rule used in (1) above is

```
(FFPROM                           ; Rule for "fulfilling promises"
  ((NO-EXIT P NIL)                ; Entry point is P for NO-EXIT use
    (RESULT P W >))               ; (Rule can also be used to infer results)
  (P PROMISE1 (SUBJ S) (OBJE D))  ; S promises an action D ...
  (S *HUM)
  (D ACT (SUBJ S))                ; ...where S will perform D.
  (W WANT (SUBJ S) (OBJE D)))     ; irrelevant to NO-EXIT use.
```

When this rule is matched with the "promised" assertion, the variable S is bound to the TMN nodes for both "John" and "he", which are therefore bound to one another in a chain.

Sometimes, especially where NO-EXIT rules are involved, a chain serves not to bind an anaphor to a particular candidate but to restrict the range of plausible candidates for an anaphor. If the text (8-2) continues

[3] They went to the restaurant.

then inference is needed to decide whether John and Mary or the monkeys went to the restaurant. From [3] SPAR infers (using mechanisms described later in this chapter) that "they" went to the restaurant because they wanted to eat. However, this inference relies on "they" being human and not animal. There is a NO-EXIT rule to the effect that "X wants Y to eat or drink something" is highly plausible if X and Y cospecify; in other words, that the goals of eating and drinking can arise spontaneously without any need for further explanation based on the story context. The resulting chain binds "they" not directly to either John and Mary or the monkeys, but rather to an unidentified human (or group of humans). This binding, when compared with the predictions of the AR rules, enables the arbitrator, the component of SPAR that coordinates the predictions of different knowledge sources, to rule out the monkeys.

8.2.4 Negative chains

All the inference chains we have considered up till now have been positive; that is, they result from the inference component deciding that two inferences confirm one another, and that therefore certain pairs of anaphors and antecedents should be identified. SPAR will accept these identifications unless they are overridden by other, stronger predictions. However, it can happen that two inferences at the ends of tendrils potentially contradict one another, or match negatively; that is, they are incompatible if corresponding elements in them are identified. But such matches still lead to chains, in the manner detailed below. Sometimes, a negative match leads to a negative chain, which, rather than predicting that the entities it binds should be identified with one another, predicts that they should ***not*** be (or, if there is more than one pair, that they should not all be). At other times, a negative match leads to a positive chain of the kind we have discussed up till now.

A simple case of a negative match leading to the formation of a negative chain is in

(8-6) [1] John killed Bill.
[2] He ran away.

where the AR rules predict that "he" is either John or Bill, and CSI is invoked to choose between them. CSI infers, as an analytic consequence of (i.e. an extraction from) [1], that Bill did not subsequently perform any deliberate actions; this would preclude his running away. Comparing this extraction from [1] with the representation of [2] gives a negative match in which "he" is bound to Bill. A negative chain binding "he" to Bill is therefore returned. Its effect is to predict that whoever "he" refers to, it is not Bill. The arbitrator notices that this chain clashes with the AR rules' prediction that "he" is Bill; the alternative prediction, that "he" is John, is therefore accepted.

In order to incorporate negative matches and negative chains into our inference mechanism, we have to decide both what constitutes a negative match, and also, it turns out, under what circumstances a negative match justifies forming a positive chain rather than the more obvious negative one. We will tackle these two problems in order.

Under what circumstances do two inferences match negatively? One circumstance, as just exemplified in the match between "Bill ***did not*** perform any actions" and "he ran away", is where they would match positively but for the fact that one of them is negated. Another is where, assuming the appropriate bindings, they represent incompatible propositions about the same entity. The range of such incompatibilities is wide; SPAR is limited to assuming the uniqueness at any one time of the state of an object (as described by its adjectival modifiers), its location and its ownership.

Secondly, what sort of chain (if any) should be formed when a negative match is found? The negative match in (8-6) resulted in a negative chain being constructed; however, there are situations in which the formation of a positive chain seems a more appropriate response to a negative match. For the text

(8-7) [1] The soldiers fired at the women.
[2] They were not hurt.

one might reason as follows: a possible consequence of [1] (expressed by a CSI) is that the women were hit, and a possible consequence of that is that they were hurt. Comparing this inference with [2] would give rise to a negative match in which "they" is bound to the women. However, this binding is (according to my intuitions at least) the correct one. In contrast to (8-6), the negativity of the match indicates not that the binding is wrong but that sentence [2] describes an exception to the normal course of events. In other words, a positive and not a negative chain is appropriate here. In general, then, how can we decide when a negative match should lead to a positive chain rather than a negative one?

On this problem, Wilks [1975b] wrote:

> "My speculation...is that there are reasonably well defined circumstances of match where negation is vital, and where two template forms that differ by a negation element ***cannot possibly*** match. However, there are other circumstances...where a template and its negated form can fuzzy-match, if no preferred non-negated form can be found. In such circumstances a negated form is a better relevance match than nothing." (Wilks' italics).

We will now attempt to delimit the "reasonably well defined circumstances" to which Wilks refers.

We first observe that a negative match must indicate one of the following.

(a) The current sentence describes an unexpected turn of events, as in (8-7), where we expected the women to be hurt (i.e. we inferred from [1] that they would be), but they were not. In such cases, what actually transpires is the negation of what we expect. One of our ***inferences*** is incorrect (i.e. its negation is true), but the ***bindings*** produced by the match are in fact correct, and therefore we should construct a ***positive*** chain to encourage SPAR to accept the bindings.

(b) The inferences at the ends of both tendrils are correct (i.e. they are true of the story) so the bindings that would make them contradict each other cannot all be correct (as in (8-6), where "he" could not be Bill because Bill was dead). Therefore we should construct a ***negative*** chain.

(c) The inferences at the ends of both tendrils are correct, but the relevant bindings are not in fact ruled out, because the inferences are true at different times or in different ways, etc. That is, the assumption involved in completing the chain - that the two inferences describe the same situation - is unwarranted. Therefore we should not construct a chain at all.

When faced with a negative match, which of the explanations (a), (b) and (c) should we prefer? Explanation (c) has already been dealt with for positive matches by the "non-vacuousness" criterion (section 8.2.2); there is no obvious reason to demand a closer negative match than we would a positive one.

Assuming, then, that the non-vacuousness criterion is met, we need to be able to choose between (a) and (b). If the tendrils contain only extractions, then (a) is impossible, because extractions cannot be wrong. However, if CSIs are involved, then (a) seems preferable. This may be because (a) leads to an interpretation that makes the current sentence both more relevant to the hearer's expectations (as simulated by the inference process) and more informative. Thus in (8-7), if we accept (a) and interpret [2] as asserting that the women were not hurt, then [2] is highly relevant to the expectations set up by [1]. In contrast, if we accept (b) and interpret [2] as saying that the soldiers were not hurt, then [2] is irrelevant to our expectations, and indeed almost contentless; having read [1], we do not expect to hear about the condition of the soldiers. In other words, (a) seems preferable (when it is plausible at all in terms of the events described) because it leads to a more coherent interpretation of the story.

Therefore if either tendril contains a CSI, SPAR accepts (a) and creates a positive chain; if not, then (a) is impossible, so it accepts (b) and creates a negative chain.[3]

8.3 Constraining inference

One of the most difficult problems to be overcome in a system that understands stories (or any other kind of text) is that of constraining the course of inference so as to generate, as far as possible, only inferences which are both plausible and relevant. This problem is especially severe in systems whose inference is mainly local in range. A number of partial solutions have been developed: creating a taxonomy of inference types to limit the kinds of inferences that could be generated in particular cases and/or combined together (Rieger's MEMORY, Cater's AD-HAC and, on paper, Wilks' PS); interest and certainty ratings (Rieger and Cater); applying saliency measures to rule clusters (Hobbs); putting a fixed upper bound on the length of an inference path (Wilks); invoking inference only when a specific ambiguity needs to be resolved (Wilks); and using focus to constrain inference (Sidner, proposed).

This section describes how inference is constrained in SPAR in order to enhance both ***accuracy*** (i.e. correctness) and ***efficiency***. Accuracy is increased by means of a development of Wilks' [1977b] taxonomy of inference types, which helps to ensure that the inferences made are plausible ones, while efficiency is gained by adopting Sidner's proposal that focus should be used to limit the inference process. Using focus in this way helps to ensure that only relevant inferences are made: "relevant" in the sense of being likely to lead to undetermined anaphors being resolved.

---

[3]These preferences can be influenced by adverbs and conjunctions connecting the two sentences (although the grammar of Boguraev's analyser, and hence SPAR, do not cover them). An adversative (Halliday and Hasan [1976]) connector such as "but" or "although" biases us towards explanation (a); in

(8-8) John killed Bill, but he ran away.

the "but" suggests that the laws of nature have momentarily been suspended, and that even our analytic inference (that Bill subsequently performed no actions) is incorrect. Conversely, a causal connector like "so" or "therefore" biases us towards (b); in

(8-9) The soldiers fired at the women, so they were not hurt.

the "so" suggests that the clause that follows it is not abnormal, so that "they" is not the women; we interpret the second clause as meaning that the soldiers were not hurt.

8.3.1 A taxonomy of inference types

In a paper written after development of the PS system ceased, Wilks [1977b] looked at the nature of causal explanations for events, particularly the explanations represented by the chains which PS's CSI component constructed to resolve pronouns. Working from the assumption that "every structure we propose, and every coding of the notion of causality, must be related in principle to concrete needs of language analysis" (p238), Wilks suggested that only two types of causal connection between events (as compared to at least sixteen in Rieger's MEMORY) needed to be distinguished. When the deliberate action of an animate agent needed to be explained (to resolve a pronoun), an explanation in terms of the agent's goals should be sought; whereas for a non-deliberate event or state, an explanation of cause (in terms of physical and not mental processes) was to be preferred. Causal CSI rules were to be categorised, in both directions of application, as either GOAL or CAUSE type. A third, non-causal type of rule, expressing ideas like "If A is a part of B, look for B not being a part of A", was categorised as IMPLIC, but the function of IMPLIC rules in the overall inference scheme, and their place (if any) in the implemented PS system, was not discussed.

An extended version of Wilks' GOAL/CAUSE taxonomy is used in SPAR for both extraction rules and CSI rules, and helps to ensure that as far as possible, the inferences made, and hence the chains built from them, are plausible ones.

Because Wilks' choice of names for his inference types is potentially confusing,[4] we will use the terms RESULTS-FROM instead of CAUSE and MOTIVATED-BY instead of GOAL. These two types have inverse types RESULT and MOTIVATES respectively; inverses are needed because the direction of application of a causal rule will depend on whether we are inferring forwards in (story-world) time from causes to effects or backwards from effects to causes. (As we saw earlier, inference in only one direction is insufficient). The forwards/backwards distinction is a vital one; if it is ignored, we run the risk of forming fallacious inference chains. For example, for

(8-10) [1] John gave Bill a banana.
[2] He ate it.

we might reason that a precondition of [1] is that John has the banana, and a precondition of [2] is that "he" has "it", and then mistakenly match the two

---

[4]Specifically:

(1) It is not immediately obvious from the name that a "cause inference" from an event or state A is a possible cause of A rather than something that A possibly causes.

(2) Whereas we can explain an event with a sentence like "the <u>cause</u> of the rock falling off the cliff was John pushing it", we cannot, as Wilks' choice of terms might imply, analogously explain an action by saying "the <u>goal</u> of John eating a sandwich was that he was hungry".

(3) Both CAUSE and GOAL are already names of semantic primitives, the first an action and the second a case.

(4) Since both CAUSE and GOAL are types of *causal* inference, it is misleading to use the term CAUSE for only one of them.

preconditions.

The stories processed by SPAR are constrained to chronological order; thus if one event is described before another, SPAR assumes it occurred earlier. (At the end of this subsection we will look more closely at the implications of this assumption). Chronological order in stories implies that, except in certain quite rare cases of cataphora which SPAR ignores, candidate starting points are always chronologically prior to anaphor ones. SPAR therefore always reasons forwards in (story-world) time from candidate starting points and backwards from anaphor starting points. (However, some of SPAR's inference rules are non-temporal; such rules, which we will examine below, are applied in the same way in both backward and forward inference).

The following possible inferences illustrate informally both types of inference (each with its inverse) that we have postulated so far.

"The rock fell off the cliff"	RESULTS-FROM	"John pushed the rock".
"John pushed the rock"	RESULT	"The rock fell off the cliff".
"John ate a sandwich"	MOTIVATED-BY	"John was hungry".
"John was hungry"	MOTIVATES	"John ate a sandwich".

These inference types are exactly equivalent to Wilks' CAUSE and GOAL types, differing from them only in name and in making explicit the forwards/backwards distinction. Such a taxonomy does seem, as Wilks intended, to cater for those inference rules that can be used to provide causal *explanations* of events.

However, not all inference rules useful for anaphor resolution are explanatory in the sense of giving a satisfactory account of *why* something happened. (See Wilensky [1981,1983] for a detailed theory of explanation in story understanding). For the text

(8-11) Fred gave John a hammer. <u>He</u> hit Bill with it.

the underlined pronoun can be resolved by reasoning (a) that Fred giving John the hammer resulted in John having the hammer; (b) that whoever hit Bill with "it" must have had "it"; and therefore (c) it was John who hit Bill with the hammer. Such reasoning is sufficient to resolve the pronoun, and is in some broad sense causal, but it does not really explain *why* John hit Bill, i.e. what his motives were.

The inference mechanism must not treat such broadly causal but non-explanatory "enablement" rules in the same way that it treats its RESULTS-FROM or MOTIVATED-BY rules. Whereas, according to Wilks' principle, when presented with a deliberate action we should ask what it is motivated by (i.e. apply MOTIVATED-BY rules), and when presented with a non-deliberate event or state we should ask what it results from, enablement inferences are appropriate for both categories. In (8-11), an enablement inference helped us to resolve a pronoun in a sentence describing a (presumably) deliberate action; however, a similar inference would be equally helpful in a non-deliberate case such as

(8-12) Fred gave John a hammer. <u>He</u> dropped it on his big toe.

We therefore extend our taxonomy to include rules of type ENABLES (temporally forwards) and ENABLED-BY (temporally backwards).

One further category of rule is also necessary. As we saw earlier, SPAR resolves the "his" in

(8-13) John promised Bill that he would mend <u>his</u> car.

by reasoning that what Bill is promised is likely to be something he can be expected to want, and that people generally want their own possessions to be in working order. The first rule applied here is not causal, at least not in any straightforward sense; although there may be some causal relationship (in either direction) between John promising to mend the car and Bill wanting it mended, the applicability of the rule is based not on any such relationship but on an aspect of the meaning of the word "promise" (as opposed to, say, "threaten"). Such rules, which represent associative rather than straightforwardly causal relationships, are classified in SPAR under the inference type SUGGESTS; they are used in the same way in both forward and backward reasoning.

Figure 8.1 summarises SPAR's inference rule taxonomy and the situations in which those rules are applied. The taxonomy includes not only the types discussed in this section, which derive one inference from another, but also the rather different NO-EXIT type, discussed in 8.2.3, which constructs a chain from a single tendril.

It should be emphasised that although the taxonomy is partly ***motivated*** by the intuitive differences between the relationships characterised informally in the first column, it is ***justified*** by the differences in mode of application described in the others. Each type of inference rule is used differently in at least one of the following respects: (1) whether it is used to extend tendrils or directly to complete chains; (2) whether it is sensitive to temporal direction; (3) whether, in backward inference, it is applicable to actions, non-actions or both. Thus although SPAR's taxonomy is more complex than Wilks', it is still, as Wilks [1977b] proposed, related to the concrete needs of language analysis.

It appears that further work along these lines would reveal the need for a still larger taxonomy. Cater [1981] argues for a taxonomy of eleven types of inference, each with its own inverse, largely on the grounds that different types are affected differently by the propositions to which they are applied being negated and/or involving "ability" (i.e. the modal verb "can"). Neither negation nor ability have been thoroughly investigated in the present research; however, Cater's experience suggests that, if they were, further inference type distinctions would become necessary. Thus Wilks' argument that all type distinctions should correspond to processing differences, though convincing, does not seem to imply that the size of the taxonomy will be as small as Wilks concluded, or even as small as SPAR's. However, a comparison of SPAR's inference mechanism (including the extensions to be described in 8.4 below) with those of Cater [1981] and Rieger [1975] suggests that the ranges of inferences that each system could in principle perform are roughly coextensive; therefore developing SPAR's taxonomy would principally involve making finer distinctions between existing kinds of inference rules rather than introducing altogether new kinds of rules.

We conclude our discussion of inference rule types by looking more closely at the chronological order restriction and arguing that although it makes the inference process simpler, the techniques described in this chapter would still be valid (though naturally less complete) if it were removed.

Relationship characterised	Rule type for forwards inference	Rule type for backwards inference	In backwards inference, applies to actions, non-actions or both?
No relationship; plausibility of single pattern only	NO-EXIT	NO-EXIT	Both
Associative (not straight-forwardly causal)	SUGGESTS	SUGGESTS	Both
Enablement (broadly causal but not explanatory)	ENABLES	ENABLED-BY	Both
Motivational (explanatory, relating mental state to action)	MOTIVATES	MOTIVATED-BY	Actions only
Resultative (explanatory, relating non-action to its cause	RESULT	RESULTS-FROM	Non-actions only

Figure 8.1: SPAR's taxonomy of inference rule types

If the restriction were removed, the temporal relationship of an anaphor starting point and a candidate one could often be deduced from such factors as tense, aspect, and whether the starting points described discrete events or continuous states. For example, if two discrete events are described in simple past tense in successive sentences without temporal modifiers, chronological order is virtually certain. However, in cases where the temporal order was unclear, it would be necessary to infer in both temporal directions from both anaphor and candidate starting points. Even so, the forwards/backwards distinction would still be crucial; it would be wrong to mix forwards and backwards inferences in the same tendril, or to form a chain from two forward tendrils or from two backward ones. The inference mechanism would effectively have to do twice as much work (one invocation for each possible event ordering), and the arbitrator would have to make sure that the temporal relationships implicit in the chains it accepted were consistent (e.g. if one chain presupposed A occurring before B, and another chain presupposed the opposite, they should not both be accepted).

Thus to relax the chronological order constraint, we would need (a) to recognise when the temporal ordering of two starting points was potentially ambiguous; (b) in cases of such ambiguity, "double up" the present mechanism by inferring in both directions from both starting points; and (c) ensure that the temporal relationships presupposed by accepted chains were consistent. To implement an approximation to such capabilities would not appear to present any overwhelming difficulties, and would in any case only involve augmenting the procedures described here and not replacing them. Therefore while the chronological order constraint makes SPAR's task easier, the techniques presented here would still be valid (though more complex in application) if it were removed.

8.3.2 Using focussing to constrain inference

Having looked at SPAR's partial solution to the first aspect of the problem of constraining inference - that of generating only plausible inferences - we now address the second aspect: ensuring that the inferences made are relevant to anaphor resolution, i.e. are about appropriate entities. As we saw earlier, Sidner claimed that the predictions of her anaphor interpretation rules could be used to constrain inference; however, she did not specify in detail how this could be done for "special mode" inference (the inference required to choose between several almost equally focussed candidate specifications, rather than to assess a single one), and her claim has apparently not previously been tested for such inference, at least in a story processing system. However, SPAR's behaviour tests, and in fact supports, Sidner's claim.

We saw in chapter 6 that when SPAR's inference mechanism is invoked, it is handed a list of still-undetermined anaphors and a list of their candidate specifications as determined by the AR rules, word-sense-oriented semantic matching and configurational constraints. These lists of what we will call <u>active</u> anaphors and candidates are used in essentially the same way to achieve three different ends: to select appropriate starting points for inference; to "prune" tendrils to prevent them going off in directions which are unlikely to be fruitful; and to terminate inference if and when it has produced enough plausible chains. The fact that the results of focussing (and other processes) can be used in this way is significant not only because it constitutes evidence for Sidner's claim that focussing can be used to constrain inference, but also because it is especially important to constrain inference in a shallow processing environment where inference has to be as simple and as limited as possible.

The three ways in which the lists of active anaphors and candidates are used to constrain inference will now be described.

(1) <u>Selecting starting points</u>. As was argued in 7.4.2, it is important to start inference only from what is actually ***asserted*** in the text: that is, from propositions not occurring in intensional contexts. For example, the sentence

(8-14a) The soldiers fired at the women.

asserts that a firing took place; but

(8-14b) The officer ordered the soldiers to fire at the women

does not, since the soldiers might not have obeyed. In the latter case, we may still want to make inferences about the possible effects of the soldiers firing, but such inferences must be based on the assumption (itself a CSI) that the soldiers obeyed the order; i.e. the firing action is not a valid ***starting*** point for inference, but should be inferred from such a starting point, thus increasing by one the length of any resulting tendrils.

In general, even if non-assertions are ruled out, we may still have a large number of potential starting points for inference, because any of the information asserted in a text may be relevant to making inferences useful for anaphor resolution. However, this is not true in the specific case of Wilksian inference. Because a Wilksian extraction rule or CSI rule is local, its output cannot mention any TMN element - and ***a fortiori*** any active anaphor or candidate - which is not also mentioned in its input. Since inferences not mentioning active anaphors and candidates cannot give rise to useful chains (i.e. chains which bind at least one active anaphor to an active candidate) there is no reason to take as a starting point any assertion not involving an active entity. (Hobbs [1976], in his semantic approach to pronoun resolution, used the suggestions of his syntactic algorithm similarly to select starting points).

Given these two constraints (the elimination both of non-assertions and of assertions not involving active entities), selecting the set of ***anaphor*** starting points is straightforward, since the single current sentence in which the anaphors occur will contain only a few assertions. Any of these assertions which involve an active anaphor therefore qualify as starting points. On the other hand, there are potentially many ***candidate*** starting points in the context, even when the two constraints are applied to these, because an active candidate may have been involved in many previous assertions. SPAR therefore selects a handful (typically three to five) of assertions representing either states (which may still hold) or recent events (which, although finished, may be relevant to what is described in the current sentence). Because only a few of the assertions in the context are selected, the amount of inference done is much reduced. The exact heuristics for selecting candidate starting points were arrived at on a trial and error basis, and their specific detail is of no special importance; however, they normally succeed in picking out all the assertions necessary for inference and not too many irrelevant ones. When they fail to do this, or indeed when an error of any other kind is made during inference, incorrect chains may be constructed or correct ones may be missed; but in such cases, as we saw in 6.4.2, other components of the system usually produce enough correct predictions to ensure that anaphors are eventually resolved correctly.

(2) Constraining the course of inference. Just as all anaphor and candidate starting points are assertions mentioning active anaphors and candidates, so SPAR only retains an inference if that inference mentions an active entity. If it does not, then it cannot lead to any useful chains; and, because of the strictly local nature of the inference rules, neither can any further inferences derived from it. For example, in the third sentence of the story beginning

(8-15) [1] John promised Bill that he would mend his car.
[2] He took it to his friend's garage.
[3] He tried to persuade his friend that he should lend him some tools.

the AR rules determine that possible specifications (i.e. active candidates) for both of the underlined pronouns are John and John's friend, but not Bill (who has drifted out of focus). The inference mechanism is then invoked to resolve these pronouns. During forward inference, SPAR infers from [1] that Bill wants John to mend the car, and then that Bill wants the car to function (a consequence of mending it). However, since this second inference does not mention either of the active candidates (John and John's friend) it is thrown away as soon as it is made, because it cannot contribute to any useful chain.

(3) Terminating inference. The sets of active anaphors and candidates can bé used to constrain not only the starting points and course of inference but also its conclusions. We have seen that Wilks envisaged inference terminating when all "problem pronouns" (in our terms, "active anaphors") had been bound by a completed chain, or when all possible inferences up to a preset maximum length had been made, whichever occurred sooner.

A modification to this principle is required (though only approximately implemented, as described below) in SPAR. Because of the way completed chains are incorporated into the overall anaphor resolution process, there will be occasions on which inference can be stopped before every active anaphor has been bound by a chain; and there will be other occasions when inference should not be stopped even though every active anaphor has been bound.

A possible occasion of the first type is as follows. In

(8-16) [1] John was in a foul mood.
[2] He saw Bill.
[3] He shouted at him.

inference will be required for [3], with active anaphors "he" and "him" and active candidates John and Bill. If SPAR had a CSI rule to the effect that a possible reason for shouting is being in a bad mood, it would be able to form a chain binding "he" to John. Although the chain would not bind "him", it would in fact enable the arbitrator to resolve both anaphors, because the MGC constraint would prevent "he" and "him" cospecifying.

Conversely, binding every active anaphor is not always sufficient grounds for terminating inference, since the arbitrator may later reject some of the chains involved as incompatible with the predictions of the AR rules, with configurational constraints, or with shorter chains. If this happens, any anaphors which could have been resolved by further inference will not be.

These two situations can only be reliably recognised by the arbitrator. The most efficient way to proceed would therefore be to make the inference mechanism *suspend* (not terminate) whenever a chain (or chains) was formed which was not obviously implausible (for example by breaking a configurational constraint or binding an active anaphor to an inactive candidate). If the arbitrator then concluded that all anaphors were finally resolved, no more inference would be needed; if it did not, inference would be restarted, looking for chains one step longer.

SPAR implements an approximation to this behaviour by rejecting chains which break a configurational constraint or which conflict with the AR rules by binding an active anaphor to an inactive candidate. It also rejects as useless chains which do not bind active anaphors at all. As we saw earlier, inference terminates as soon as every anaphor is bound by an unrejected chain or when the maximum chain length has been reached. In practice this usually means that the right amount of inference is done.

8.4 Extending the basic mechanism

In this chapter, we have until now accepted Wilks' inference ideas fairly uncritically. We have concentrated on fleshing them out and on making the minimum of alterations necessitated by the differences between PS and SPAR in areas such as representation and the role of focus. However, one possible objection to Wilks' model of inference is that, like Rieger's and Charniak's early theories, its restriction to only local inference and mainly causal knowledge seriously limits the kind of reasoning we can expect it to perform. In this section we will see how three further kinds of reasoning are built into SPAR's inference mechanism. These additions significantly extend the mechanism described so far but retain the concepts and apparatus that make Wilks' theory of inference attractive. The additions involve exploiting, respectively, (1) knowledge about the stereotyped behaviours associated with specific objects and places, which has often been represented using scripts; (2) knowledge about how the goals of an actor relate to his actions and to one another; and (3) very weak rules which can act as a backup when the normal rules fail, as they often will in a shallow processing framework, to bind all the unresolved anaphors.

The use of these kinds of inference in SPAR, integrated with the inference methods discussed up till now, suggests that a chain-based mechanism such as Wilks' may be capable of detecting a rather wider variety of connections between events than those detected by Wilks' PS system. However, in accordance with the shallow processing approach, none of the three extensions requires significant extra knowledge to be added to the system. They are integrated with the rest of the inference process in such a way as not to alter its essentially uniform (as opposed to heterogenous) character. In particular, the extensions are orthogonal to the taxonomy of inference types developed earlier, in that they provide new ways of making inferences of those types, rather than adding to the number of types.

8.4.1 Object-oriented inference

In order to perform the kind of object-related, stereotyped inference often achieved by means of scripts (e.g. Cullingford [1978]) or frames (Charniak [1978]), SPAR accesses the information in formulas for noun senses to reason about possible events connected with objects described by those noun senses. Such object-related knowledge is normally present only in these formulas, since little detailed *world* knowledge is available to a shallow processing system. Of course, the knowledge available in a formula will typically be far less extensive than that in a script; for example, while a script in SAM could contain over a hundred event patterns (Cullingford [1981]), a formula in SPAR normally mentions at most only the most central action or two associated with a noun sense. Nevertheless, this central information is often all that is required for our purposes.

SPAR exploits object-related knowledge by means of a special mechanism which, like a CSI rule, generates a new inference from an old one. Once generated, the new inference is treated just like any other; it becomes part of a tendril, and may form part of a completed chain. Thus whereas some story or text understanding systems that have used scripts (or other knowledge structures linking sets of events together) have not made significant use of other inference methods (e.g. SAM, FRUMP (DeJong [1979]), IPP (Lebowitz [1980]); but contrast BORIS (Dyer [1982])), in SPAR object-oriented inference is fully integrated into the overall inference engine, and can be freely mixed with other kinds of inference.

The main idea behind SPAR's object-oriented inference can be stated as follows:

> If an object O has a formula which defines it in terms of a generic event or action E which has precondition P involving O, then it is sensible to infer E from P as a common sense inference (of type ENABLES).

That is, if a precondition for an event associated with an object is satisfied, it is sensible to infer that the event will occur. For example, if "restaurant" is defined (using the intermediate-style formula for clarity) by

```
RESTAURANT1: (PLACE2 (0)
                  (WHICH
                     (EAT1 (1)
                        (TYPE (GEN))
                        (SUBJ (MAN (2)))
                        (IN (PLACE2 (0))))))
```

("a place in which people GENerically (i.e. habitually) eat") then SPAR can establish by extraction that a precondition of the EAT1 action taking place is that the MAN should be located IN the restaurant. This relationship intuitively seems reversible: if it is asserted or inferred that a particular person is in a restaurant, we may want to infer that that person will eat there. Similarly, if a tool is defined as an instrument with which people mend things, and if we are told that someone has some tools (a precondition for using them to mend something), we may want to infer that they will mend something with those tools.

This form of inference is implemented in SPAR using the extraction rules which are part of the normal inference process to establish the appropriate preconditions. It works as follows.

Suppose that, during forward inference (i.e. inference from candidate starting points) SPAR comes across an inference involving a RESTAURANT1: for example (in simplified notation)

(8-17) (BE (SUBJ JOHN1-1) (IN RESTAURANT1-1))

(i.e. John is in a particular restaurant). It looks for any action or event subformulas in the formula for RESTAURANT1 (as represented in the word sense network) and finds the one headed EAT1. It then constructs, from the EAT1 subformula, a provisional inference representing the supposition that some person eats in the restaurant: say

(8-18) (EAT1 (SUBJ MAN-23) (IN RESTAURANT1-1))

where MAN-23 is a newly-generated token. SPAR then applies ***backward*** extraction rules to this provisional inference to find out what the preconditions of such an eating event would be. If any of the resulting extractions matches the original inference involving the RESTAURANT1, SPAR adds the EAT1 inference (with suitable substitutions), as a CSI, to the tendril containing the original inference. One backward extraction from (8-18) is

(8-19) (BE (SUBJ MAN-23) (IN RESTAURANT1-1))

The match with the original inference (8-17) succeeds, giving a single binding MAN-23=JOHN1-1. This binding is substituted into the provisional inference (8-18) to give the common sense inference

(8-20) (EAT1 (SUBJ JOHN1-1) (IN RESTAURANT1-1))

which is added to the tendril headed by (8-17) just as if it had been derived by a straightforward CSI rule.

One problem encountered by script-based story understanders is that of when a script should be invoked. While it might be sensible to invoke a "restaurant-meal" script on hearing that "John went to the restaurant", it would not be sensible to do so on hearing "John drove past a restaurant". As just described, SPAR tackles this problem by only making an object-oriented inference when there is a match with an extraction. Thus while "John went to the restaurant" will give rise to a RESULTS-IN extraction "John was at the restaurant" which in turn will generate an object-oriented inference "John will eat in the restaurant" in the way described above, nothing comparable will be derived from an event like "John drove past a restaurant", because that event does not in any simple way imply that anyone in particular was in the restaurant.

The analogy between SPAR's object-oriented inference and script-based story understanding yields a further interesting perspective on the problems of script activation and interaction. If Wilks' heuristic of preferring shorter chains to longer is reliable (and of course this remains unproven), then there is no need to decide separately when a "script" should be activated (i.e. whether an object-oriented inference should be made), or to worry about competition between scripts. This is because tendrils which do not lead to completed chains have no effect on subsequent processing, so a script will effectively only turn out to be "active" if it contributes to a tendril which enables a chain to be constructed; and competition between two scripts which contribute to different chains will be resolved by preferring the shorter chain. Thus making an object-oriented inference does not ***in itself*** constitute activating a script.

The mechanism could be further prevented from occasionally making implausible inferences by ruling that the elements substituted into an object-oriented inference (in our example, JOHN1-1) must not have any close prior connection to the object giving rise to the inference. Whereas it is plausible to infer from "John went to the restaurant" that John will eat there, it is much less plausible to infer from "The waiter went to the restaurant" that the waiter will eat there, because a waiter has a special relationship to a restaurant which does not involve eating. Although this constraint of not making object-oriented inferences in the presence of such special relationships is not yet implemented in SPAR, there seems no reason why it should not be.

The most obvious limitation of SPAR's object-oriented reasoning capability is that, because of its shallow processing approach, it has access to far less information than a script applier such as SAM. Whereas, say, a SAM-style restaurant script will mention many events, such as entering, sitting, ordering, eating, paying and leaving, a formula for a noun like "restaurant" will normally only mention the most central event: in this case, eating. However, the gaps can sometimes be filled by rule-based CSI; for example, if there is a CSI rule which says that people often pay for things they are given, there may be no need to specify separately that paying is something done by the customer in a restaurant, but not by the waiter or the cook.

8.4.2 Goal-oriented inference

Like object-oriented inference, goal-oriented inference in SPAR is based on a simple observation and a consequent extension of the inference mechanism which allows a certain kind of reasoning to be incorporated. For goal-oriented inference, the observation (which in itself is far from original, but which allows a novel extension to SPAR's reasoning and turns out to be surprisingly productive) is as follows: people's goals can be explained by modelling the inferences they make about the consequences of actions and events in the world. Goal-oriented inference can therefore be viewed as the dual of normal, "event-oriented" inference in the sense that if we want to explain (by backwards inference) why someone has the goal that a particular event should occur, we should simulate (by forward inference) that person's reasoning about what the consequences of that event might be.

Such simulation can lead to the resolution of pronouns. In 8.2.2, when we looked at the inference SPAR performs to resolve the pronouns in

(8-21) John promised Bill that he would mend his car.

we in fact glossed over a goal-oriented inference: the inference that if Bill wants "he" to mend "his car" then "Bill" wants "his car" to function. Specifically, the reasoning involved is as follows.

When applied to the representation of (8-21) during backwards inference, a rule of type SUGGESTS gives the following inference (in simplified form):

```
(WANT
   (SUBJ BILL1-1)
   (OBJE
      (MEND1
         (SUBJ HE1-1)
         (OBJE (CAR1-1
                  (POSS HIS1-1))))))
```

(Bill wanted "he" to mend "his car"). When SPAR tries to make further backward inferences from this one, it sees that the head primitive is WANT; i.e. that the inference describes someone's goal. It therefore applies ***forward*** inference rules to the embedded MEND1 structure. Because MEND1 is defined in the word sense network as "CAUSE to FUNCtion", the inference (in fact an extraction) is made that the MENDing will result in the car functioning. This inference is made in the same context that contained the MEND1 pattern; thus SPAR derives the ***backward*** inference

```
(WANT
   (SUBJ BILL1-1)
   (OBJE
      (FUNC
         (SUBJ (CAR1-1
                  (POSS HIS1-1))))))
```

(Bill wanted "his car" to function) which, when subsequently handed to a NO-EXIT rule, allows HIS1-1 to be identified with BILL1-1.

In making this goal-oriented inference, SPAR effectively assumes that Bill is a rational person, and infers that if he wants some event to occur, then he will also want the predictable consequences of that event.

This simple device makes whatever knowledge SPAR has about the way events proceed in the world applicable also to reasoning about an actor's goals. No special inference rules applying exclusively to goals (and not to actual story events) are are required, although of course there are some rules that relate events to the goals that they result from and give rise to. This feature of SPAR contrasts with PAM (Wilensky [1981]) whose rule base explicitly contained, for example, the information that the goal of satisfying one's hunger might give rise to the plan of eating at a restaurant. Using a mechanism such as SPAR's, such a relationship could be inferred from the straightforwardly causal knowledge, which would presumably be required anyway, that eating (in a restaurant or elsewhere) normally results in one's hunger disappearing.

The use of the same inference rules for both goal-oriented and "event-oriented" (i.e. top level in the story world) inference relies on several assumptions which, while not always true, are arguably sufficiently reliable to be be justified in a shallow processing system. The main such assumptions are:

(1) that the actor's inference processes are similar to SPAR's, and that therefore SPAR can "model" those processes by putting itself in his position. This assumption is unlikely to be completely right, because of the severe limitations of SPAR's reasoning processes compared to those of people, and because people do not always think rationally. However, there is no reason why the errors resulting from such discrepancies should be any worse than those occurring during ordinary event-oriented inference.

(2) that the actor's knowledge and beliefs about the world of the story are the same as SPAR's. This assumption too is clearly not always true, but arguably is true often enough to justify performing goal-oriented inference in this way in a shallow processing framework. Reasoning about situations where an actor's picture of the world differs from the truth is a very hard problem.

Thus there is no reason to suppose the goal-oriented inference SPAR performs is any less accurate than event-oriented inference. While there is clearly far more to reasoning about goals and plans than is covered by SPAR, the mechanism described is surprisingly useful, accurate and flexible.

8.4.3 Weak inference rules

So far in this section we have seen how SPAR's Wilksian inference mechanism is extended to perform object-oriented and goal-oriented reasoning. Both of these types of reasoning are fully integrated into the tendril-growing and chain-forming process. We now examine a third extension to the mechanism which, rather than augmenting it in the way the other two do, provides an alternative backup procedure for use when the main mechanism fails to bind all the undetermined anaphors. In a shallow processing system, such failures can be expected to occur quite often.

When the inference methods described so far can find no chain connecting an anaphor starting point to a candidate one, SPAR applies a number of <u>weak inference rules</u> to bind anaphors to candidates. These rules are also used to form chains, but in order to be able to "trap" situations where normal CSI rules are inapplicable, they have to be widely applicable and therefore extremely general in content. Their weakness is a consequence of their generality; while they are hopefully true more often than not, they apply to such a wide variety of situations that they tend to be less reliable than ordinary inference chains, even those of maximum length. Another consequence of the weak inference rules' generality is that there is no point in combining them to form chains, either with each other or with extraction and CSI rules; if this were done, every active anaphor would be bound to virtually every active candidate, and the results would be of little use. Weak rules therefore either bind anaphors to candidates by directly comparing an anaphor starting point with a candidate one, or by examining an anaphor starting point only (in the same way as a NO-EXIT rule).

SPAR uses a weak inference rule to resolve the underlined pronoun in the following story fragment.

(8-22) [1] Mary wanted to buy Susan a present.
[2] She thought that she would like a computer.
[3] <u>She</u> went to the shop which sold them.

A more powerful story understander such as PAM might resolve the underlined "she" as "Mary" by reasoning that Mary is carrying out actions to fulfil her plan of getting a computer to give to Susan as a present. However, SPAR's inference engine resolves the pronoun by means of a weak inference rule, by reasoning that sentence [2] (and indeed [1]) mentions only Mary and not Susan at "top level" outside an intensional context, since Susan participates in the story only in the context of Mary's thoughts. (We assume that the pronouns in [2] have been resolved as "Mary" and "Susan" respectively). It is therefore likely that the top level "she" in [3] refers to Mary rather than Susan, because Susan is in some sense not present in the scene set up by [1] and [2], while the referent of "she" in [3] is present in this scene. This inference uses only linguistic information that SPAR already has (since the "context of identity" of each node has been calculated).

SPAR's weak inference mechanism is comparatively undeveloped (in fact only three weak rules are implemented), so no very specific claims can be made for it. However, something along these lines is needed to rectify the inevitable failings of the inference mechanism of a shallow processing system.

8.5 Summary and evaluation

We began this chapter by looking in some detail at the kind of world knowledge that should be encoded in a shallow processing system. It was argued that the total amount of world knowledge in the system should not greatly exceed the amount of linguistic knowledge. As a consequence of this, the knowledge encoded should have two main characteristics. It should be as general as possible (i.e. applicable to a wide range of situations); and it should be as brief and succinct as possible, with only the essentials of each fact being represented. The inference component that exploited this knowledge should be a simple, uniform one, but should be able to discriminate between the contradictory chains of reasoning that would inevitably arise. Wilks' model of inference seemed a promising basis for such a mechanism.

Each of the next three sections of the chapter described some aspect of SPAR's inference mechanism and attempted to answer a question about Wilksian inference in a Sidnerian, shallow-processing environment. The three questions, and the answers suggested, were as follows.

(1) Do Wilks' basic ideas on inference seem plausible under closer examination and more thorough testing?

> In section 8.2 we looked at how SPAR grows tendrils and joins them together to form anaphor-binding chains. We found that although Wilks' techniques did not appear to need major modification, some alterations were necessary. Firstly, extraction has to be interleaved with CSI rather than just preceding it. Secondly, not all matches between inferences justify forming a chain; we had to develop a heuristic for deciding when a match was close enough for a chain to be formed. Thirdly, a single tendril can sometimes appropriately be used to form a chain by means of a NO-EXIT rule. Fourthly, we looked at the problem of negative or contradictory matches, and formulated a rule for deciding whether the resulting chain (if any) should be positive or negative. The answer to question (1) is therefore a qualified "yes".

(2) Does Sidner's framework significantly reduce the difficulty and amount of CSI needed for anaphor resolution, as she claims? If so, how? In what additional ways can Wilksian inference be controlled so as to maintain accuracy and efficiency?

> In section 8.3 we distinguished two aspects of the problem of controlling inference: ensuring that its content was plausible, and ensuring that the right entities were inferred about. We tackled the first aspect by extending Wilks' taxonomy of causal explanations to include non-explanatory and non-causal inference rules and to make explicit the distinction between temporally forward and backward inference. We saw how SPAR uses this extended taxonomy in an attempt to ensure that the right inferences are made at the right times, so as to maintain the accuracy of the inference process.
>
> We tackled the second aspect of the control problem by using the predictions of the AR rules to constrain the way in which inference was begun, carried forward and terminated. All starting points,

inferences and chains had to involve active anaphors and candidates. This procedure made explicit the means by which, as Sidner suggested, focus can be used to constrain inference.

Thus inference should be (and is, in SPAR) controlled both in the focus-based way Sidner suggests and by enforcing constraints based on an inference rule taxonomy.

(3) How can a Wilksian inference mechanism be extended to make best use of the limited amounts of general world knowledge that are available to it in a shallow processing system? How can it take maximum advantage of the available linguistic knowledge, especially that provided by word sense formulas? In terms of the classification of inference mechanisms given in chapter 7, does extending the mechanism in these two ways make it specialised, heterogeneous or uniform?

In section 8.4 we saw how SPAR's inference mechanism extends Wilks' framework in three ways. Firstly, in accordance with the shallow processing requirement that linguistic knowledge should be used as fully as possible, information in the formulas for noun senses is used to make the kind of stereotyped, "object-oriented" inferences that some systems have made using scripts or similar structures. Secondly, SPAR reasons about an actor's goals by simulating, in a limited way, the actor's own inference processes; this makes all the inference rules used for story-level inference available to goal-oriented inference. Thirdly, if the normal inference mechanism does not bind all active anaphors, a backup mechanism using very general, very weak rules is invoked. This mechanism has not been thoroughly tested, but constitutes some indication of what can be done to compensate for the lack of extensive detailed world knowledge.

The first two of these extensions are thoroughly integrated into the basic tendril-growing mechanism; the third is not, but instead stands ready to step in when the basic mechanism fails. Thus SPAR's inference component deals with several different types of world knowledge using relatively little special machinery, and therefore qualifies as a nearly uniform system.

Appendix C consists of SPAR's own commentary on its anaphor resolution process. The commentary shows the inference chains that are made in several stories, including some of those used as examples in this chapter. Appendix C, together with appendices A and B, aim to show how far SPAR's inference mechanism and rules are robust enough to work successfully on a range of stories. Some of the stories are fairly similar as wholes, but have specific differences of anaphor resolution; these are intended to demonstrate that the inference mechanism, in tandem with other parts of the system, is sensitive enough to make the right inferences at the right times. Other stories are very different in content, and show how SPAR, unlike some earlier systems, is not restricted to processing stories of a particular type.

The problem of inference is a very complex one, and it cannot be maintained that anything approximating to a complete solution to it has been presented here. Instead, we have attempted to show that a system which lacks extensive detailed and specialised world knowledge can still exhibit flexible and surprisingly powerful inference behaviour which can make an important contribution to resolving anaphors. Whether this behaviour is flexible and powerful enough to be useful in a "real" text processing system of some kind remains to be seen; however, the evidence presented in this chapter gives some modest grounds for optimism.

9. Paraphrase generation in SPAR

> "Prose = words in their best order; poetry = the *best* words in the best order."
>
> *S.T. Coleridge, "Table Talk", 1827*

This chapter describes SPAR's English generator, which provides sentence by sentence paraphrases of the stories SPAR processes. Such paraphrases enable the reader to determine what interpretation has been assigned to each linguistic ambiguity in the input. A paraphrase of each input sentence is generated immediately after the sentence has been interpreted by SPAR, rather than after the whole text has been processed, to ensure that information from later sentences is not used in the paraphrase; such information is not available when making the interpretation decisions that the paraphrase is intended to reflect, and so it would be inappropriate to include it in the paraphrase.

To build a suitable generator, we must answer two questions:

(1) what sort of paraphrases do we want? and

(2) what techniques should we use to derive those paraphrases from pieces of SPAR's meaning representation?

The answers to these questions are interdependent: obviously, we need to know what we are trying to achieve before we can design the generator; on the other hand, and less obviously, the detailed nature of the desired paraphrases is more usefully specified in terms of the structure of the generator than in purely linguistic terms. In this chapter, we therefore progress from the general to the specific, addressing questions (1) and then (2) in turn at each level of generality.

Thus sections 9.1 and 9.2 present some general answers to (1) and (2) respectively. In 9.1 we look at the properties SPAR's paraphrases must have if they are to provide support for the claims made in this thesis, and we conclude that, first and foremost, the paraphrases must show what choices have been made in resolving ambiguities: chiefly the anaphoric ambiguities that SPAR is primarily intended to resolve, but also the lexical and structural ones that Boguraev's analyser cannot resolve in isolation.

In the light of this conclusion, in section 9.2 we answer question (2) also in general terms. This is done in two stages. Firstly, we review relevant previous work in generation, especially that of Tait (Tait and Sparck Jones [1983]) and that of McDonald [1981]. Secondly, we propose that SPAR's generator should have the following components:

(a) A strategic component which decides what information should be expressed in the paraphrase of the current sentence. This component takes the TMN (text model network) node representing the main event described by the current sentence and derives from it a Boguraev-style dependency structure containing all and only the required information.

(b) A tactical component which decides how this information should be expressed. This component, which is based on Tait's generator, turns the dependency structure into a surface syntactic tree and then into an English sentence.

Having established our general framework, in sections 9.3 and 9.4 we move on to specifics. In 9.3 we return to question (1); we develop some specific linguistic rules for ensuring as far as possible that the paraphrases satisfy the requirement established in 9.1 (that the paraphrases should make clear ambiguity resolutions) in the areas of lexical, structural and anaphoric ambiguity. These rules are expressed in terms of the generator structure outlined in 9.2. We then, in section 9.4, revert to question (2) and answer it in detail; thus we show how the specific rules established in 9.3 are embodied in SPAR's generator, illustrating the generator's operation with examples.

Finally in section 9.5 we summarise and evaluate the generator.

9.1 What should a paraphrase achieve?

In chapter 4, SPAR's meaning representation was justified not on independent grounds but by assuming the general validity of Wilks' and Boguraev's arguments for their respective representations, adapting those arguments as required. In the same way, we will begin this section by summarising Boguraev's approach to paraphrase generation, set out in detail in Boguraev [1979]. Having summarised Boguraev's approach, we will show how it is applicable, with certain important modifications, to our concerns.

Boguraev's approach is highly relevant because he was concerned, as I am, with demonstrating ambiguity resolution by means of paraphrase; other language processing systems that produce paraphrases, e.g. cooperative systems, have tend to use paraphrase as a means to an end such as query clarification or verification rather than as the final output of the system. Also, as chapter 4 showed, SPAR's representation is in many ways similar to Boguraev's, from which it is derived.

We will not try to formulate a precise definition the notion of paraphrase; as Boguraev observes, "there is no strict definition of what should be accepted as a good paraphrase of a sentence...One man's paraphrase is another man's corruption of meaning". According to Smith and Wilson [1979], "a sentence which expresses the same proposition (has the same meaning) as another sentence is a paraphrase of that sentence...Paraphrase is to sentences as synonymy is to lexical items". However, just as exact synonymy between lexical items is very rare, so adopting a definition such as Smith and Wilson's will not allow us to generate a range of paraphrases wide enough to provide convincing evidence of the system's capabilities. We will therefore operate with a relatively informal notion of paraphrase, and aim to produce paraphrases which, while they may not always have exactly the same meaning as the input, are related to it in such a way as to demonstrate that appropriate interpretation decisions have been made.

9.1.1 Boguraev's view of paraphrase

Boguraev intended the paraphrases his generator produced to achieve two main purposes: firstly, to allow a more reliable evaluation of the meaning structures produced by the analyser than inspecting them directly would provide, and secondly, to demonstrate that the structures possessed three properties, which we will label (B1)-(B3) for future reference:

(B1) they represented correct interpretations;

(B2) they were sufficiently coherent and informative for a generator to produce paraphrases from them without access to the original input sentence; and

(B3) they were far enough removed from the input sentence to achieve a degree of normalisation that enabled paraphrases with a variety of word orders to be generated.

In order to achieve these aims, a paraphrase should be a grammatical sentence that reflects the nature of the dependency structure from which it was derived; however, it must differ from the input sentence enough to show what choices had been made in resolving lexical and structural ambiguities and also to demonstrate that the representation is not too dependent on the input sentence's word order. Boguraev notes that "paraphrasing a sentence by itself, however genuine the processing involved, is not very convincing". (SPAR's generator does sometimes produce a paraphrase identical to the input sentence, but only when the input contains no ambiguities, i.e. when SPAR's interpretative task is comparatively trivial. In such cases, the interesting ambiguity problems are elsewhere in the text.)

A paraphrase can differ from the original sentence in many ways, ranging from simple word replacement:

"John struck Mary" → "John hit Mary"

to extensive reformulation involving explicit and perhaps complex inference or summarising:

"The President declared that he could not commit himself to endorsing ..." → "The President refused to endorse ..."

Boguraev argued that while, for his purposes, explicit inference was unnecessary, simple word replacement was insufficient. He argued that his generator must be capable of synonym substitution (both of words and of phrases, treated as unanalysed units) and of constituent manipulation (both reordering and internal restructuring).

9.1.2 Applying Boguraev's argument to SPAR

How far does Boguraev's argument also apply to SPAR? Since, like Boguraev's system, SPAR sets out to resolve ambiguities, its paraphrases must be different enough from the input to show what ambiguity resolution decisions have been made; and producing such paraphrases will involve synonym substitution and constituent manipulation. However, SPAR differs from Boguraev's system in two important respects: in addressing primarily anaphoric rather than lexical and structural ambiguity, and in processing multi-sentence texts rather than isolated sentences. Moreover, SPAR begins with Boguraev's analyser structures, which we will assume possess the properties that Boguraev needed to show they have. Paraphrase generation for SPAR is therefore rather different from paraphrase generation for Boguraev. It is a more extensive task (it involves deciding what to say as well as how to say it); but in some ways it is easier, since it is easier to ensure that paraphrases differ interestingly from the input.

(1) <u>SPAR primarily addresses anaphoric ambiguity</u> rather than lexical and structural ambiguity. This means that operations like synonym selection and constituent reshuffling, while sometimes necessary, are less important than ensuring that anaphors are paraphrased in such a way as to express their specifications as clearly and unambiguously as possible. This will involve including enough information to distinguish the specification of the anaphor from every other element in the TMN; in other words, before deciding (as Boguraev's generator did) how to express something in English, we have to decide exactly what information needs to be expressed. SPAR's generator will therefore need a strategic component, to decide what to say, as well as a tactical one, to decide how to say it. (As the chapter proceeds, we will specify in more detail what lies behind these rather crude characterisations of the strategic and tactical aspects of the task. Various complex rhetorical issues which are often important in generation, such as those discussed by McKeown [1985], may safely be ignored because of the generator's specialised goals.)

Although the current TMN fragment has already been integrated into its context by the time the generator is invoked, a pointer is kept to the TMN node derived from the main clause of the current sentence. When this node is handed to the generator, it is quite easy, because of the close resemblance between the TMN and the dependency structures that gave rise to it, for the generator to distinguish the content of the current sentence (which it needs to paraphrase) from the rest of the TMN. The problem of deciding "what to say" is therefore essentially one of deciding how much (and what) detail to include when indicating the specifications of resolved anaphors.

(2) <u>SPAR processes multi-sentence texts</u>. A text that is acceptable as a paraphrase of another need not correspond to it sentence-for-sentence. That is, if $S_1'...S_n'$ is to qualify as a paraphrase of $S_1...S_m$, it does not follow that every S_i' is a paraphrase of S_i, even in context. This gives us a whole range of options for generating paraphrases not available to Boguraev. However, as stated at the beginning of this chapter, if SPAR did not paraphrase texts sentence by sentence, it would be much harder for the reader to see what resolution had been made for a given ambiguity in the input.

But even though we restrict ourselves to sentence-by-sentence paraphrase, a multi-sentence text representation gives us more options that an isolated-sentence one, since it allows us to generate sentences that are only paraphrases of the original ***in context***. For example, while "John opened the car's door" does not qualify as a paraphrase of "He opened the door" in isolation, it does qualify in the context provided by the initial sentence "John walked over to his car". Thus at least in the case of sentences that contain anaphors with contextual (rather than intra-sentential) antecedents, we can be fairly sure that the paraphrase will differ significantly from the original as long as we follow the practice established in (1) above of expressing the specifications of anaphors as clearly as possible, since doing so will include contextual information in the paraphrase.

(3) SPAR accepts Boguraev's structures as input. Although SPAR does in fact have access to the original word string, and attaches it to some TMN nodes, this information is used only for debugging and clarificatory purposes and not to aid processing. The only exception to this is that it is used during resolution to establish the surface ordering of anaphors; however, this ordering information is in turn only used in a fairly minor way (see 6.3.1), and is in no sense part of the representation of the ***events*** described by the story. This means that SPAR's representation must, ***a fortiori***, be at least as normalised (i.e. independent of the form of the surface text) as Boguraev's, from which it is constructed. Therefore if we accept Boguraev's evidence for his point (B3), namely that his structures are sufficiently far removed from the input that the word ordering of paraphrases is not unnecessarily constrained, then there is no need to re-prove this point for SPAR by heavy use of transformations such as constituent reshuffling. Such transformations will only be needed to the extent required for clarifying ambiguity resolutions.

In conclusion, then, SPAR's generator will need to provide evidence that SPAR's representation has properties (B1) and (B2); (B3) is deemed already established. (B2) will be established by the successful production of grammatical sentences acceptable as paraphrases, while (B1) will be established by using synonyms to show what word senses have been selected, by such reshuffling of constituents as is needed to show what structural attachments have been made, and, most importantly, by paraphrasing anaphors in sufficient detail. The last of these actions alone will usually make the paraphrase different enough from its input to be convincing. Our aim, which will serve as the basis of the rest of the chapter, is therefore solely:

> to produce, for each input sentence in the story, an output sentence which is intuitively acceptable in the context as a paraphrase, and which shows what decisions SPAR has made in resolving ambiguities, especially anaphoric ones.

An important consequence of this is that the paraphrases produced are often rather verbose and not very readable. In particular, pronouns, being maximally ambiguous, are never used. Dale [1985] argues that normally in text generation, referring expressions should be both referentially adequate - they should contain sufficient information for the hearer to distinguish the referent from other potential referents - and efficient - they should contain no more information than is necessary for the purpose at hand. These two requirements pull against each other, ideally leading to expressions that are both unambiguous (in context) and succinct.

However, in SPAR's case, a stronger notion of referential adequacy is needed. We must aim to include enough information not just to enable the reader to pick out the intended referent, perhaps applying various kinds of knowledge to recover implicit information; rather, our description must be informative enough to ***explicitly*** distinguish the intended referent from all other possibilities. Only this will demonstrate that SPAR has indeed made the correct resolution.

Even so, the requirement of efficiency must still be applied if the paraphrases are not to become excessively clumsy. We will see in section 9.4 that this has far-reaching implications for the generator's control structure.

9.2 The general organisation of the generator

Having decided roughly what characteristics the paraphrases produced by SPAR should have, we now turn our attention to the problem of how such paraphrases can be generated. We will first survey relevant previous work in generation, and then build on that work to propose an overall organisation for SPAR's generator. This will complete our general analysis of the problems of paraphrase generation, and enable us, in the rest of the chapter, to characterise the generator more specifically.

9.2.1 Some previous work on generation

Two pieces of research into generation are particularly relevant to our concerns. Firstly, McDonald's [1981] research, which covers both the strategic and the tactical aspects of the task, is among the most important yet carried out in the area of generation; SPAR's generator in many ways resembles McDonald's MUMBLE, although the latter is far more complex and sophisticated. Secondly, Tait's English generator (Tait and Sparck Jones [1983]), which accepts Boguraev's dependency structures as input, is the basis of the tactical component of SPAR's generator.

Two further pieces of work are of secondary relevance. They are the French generator (Herskovits [1973]) used by Wilks' PS system (which was, as we have seen, was one of the main inspirations for SPAR); and Boguraev's [1979] generator which, like Tait's, constructed English sentences from Boguraev's dependency structures.

Although much more research has been done in the area of generation, it will not be reviewed here because it did not influence the design of SPAR's rather special-purpose generator, and because we are interested in generation not as a topic in its own right but as a means to evaluating the performance of the rest of the system. (However, as I will make clear at the end of the chapter, this does not mean that the generator is without interest as an independent piece of work).

McDonald's work will be described first, so that the other three systems can be discussed in the light of the comprehensive framework he provides. The other three systems will then be examined in chronological order.

(1) <u>McDonald</u> set out to address what he regarded as the central problem of generation: that of discovering "how specific utterances arise from specific communicative goals in a specific discourse context". His overall model, shown schematically in figure 9.1, consisted of an <u>expert program</u>, whose existence he assumed, and a <u>generator</u> such as his implemented MUMBLE system. We will look mainly at the model rather than the system, and will largely ignore issues of domain dependence and independence because, although McDonald stressed them, they are not of direct relevance to SPAR's generator.

The expert program would be capable of reasoning about a specific domain such as petroleum geology, but had no linguistic knowledge of its own; it was therefore entirely the job of the generator to communicate the expert program's conclusions, explanations, questions and so on to the user. The

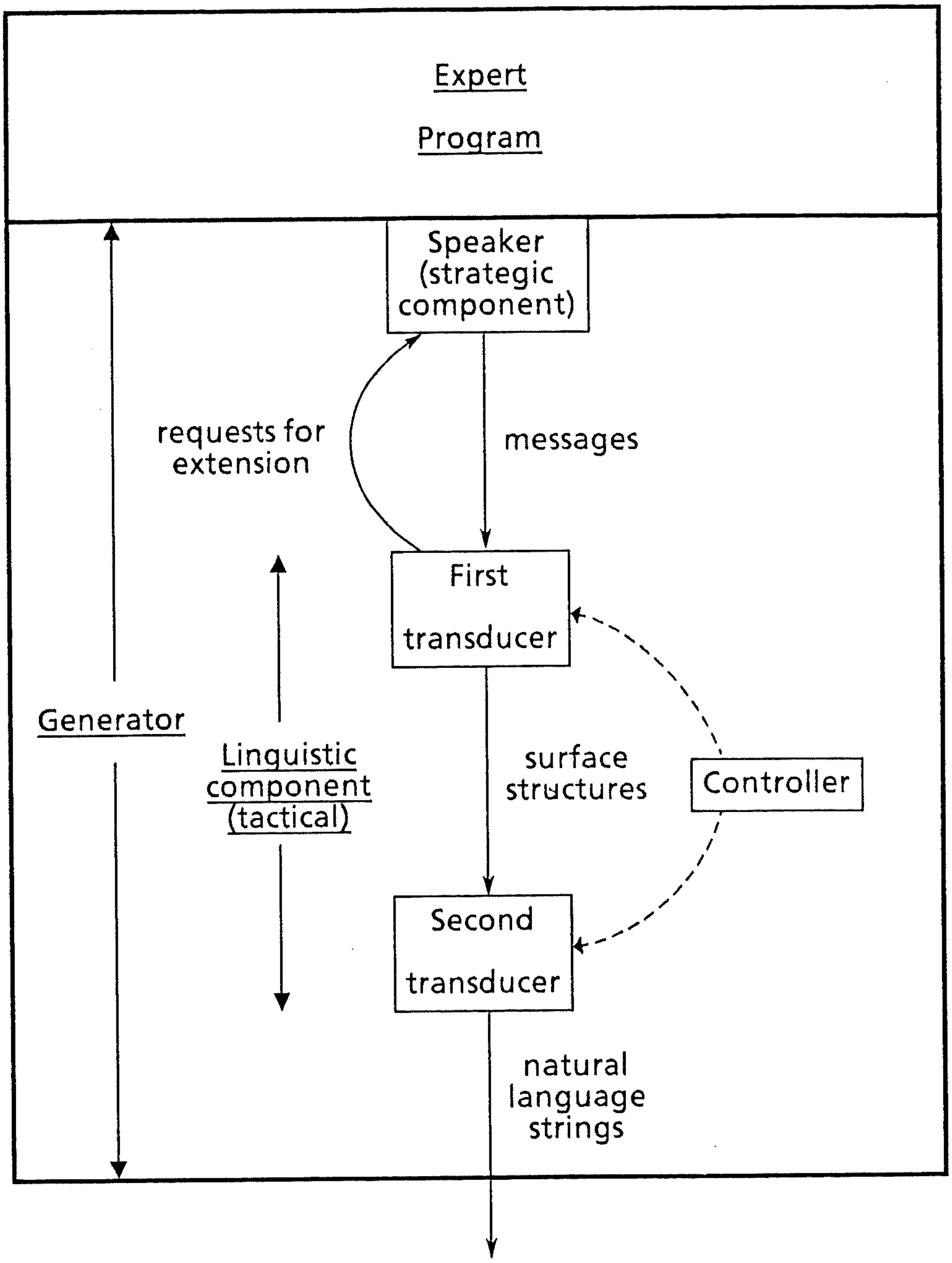


Figure 9.1: McDonald's generation model

generator consisted of a strategic speaker and a tactical linguistic component.

The speaker possessed discourse capabilities such as the ability to decide what information to include in an utterance and what level of abstraction was appropriate for an explanation. It would hand to the linguistic component messages describing the goals it wished to achieve with its utterance.

The linguistic component consisted of two cascaded transducers under the command of a single, data-directed controller. The first transducer would, when invoked by the controller, convert a part of the message into a piece of surface structure, from which the second transducer would derive a natural language utterance. Such a two-stage approach to the tactical aspect of generation is fairly common, partly because a number of operations, such as enforcing number agreement between subject and verb, are most easily carried out on the intermediate surface structure.

However, the transducers were not simply applied in series; that is, it was not the case that the first transducer converted the whole message to surface structure and then went to sleep while the second transducer converted the surface structure to an output string. Instead, both were invoked incrementally by the controller. Initially, the message was made the topmost node of the surface structure tree; the controller passed through this tree deterministically, depth-first and left to right. When the controller encountered a message element, it would call the first transducer to convert it to surface structure (perhaps with other message elements as terminals); this might involve consulting the speaker to extend the message. When the controller encountered a piece of ***bona fide*** surface structure, it handed it to the second transducer for grammatical operations, or, in the case of a terminal node, printed it after appropriate morphological conversion.

Thus McDonald's model of generation was <u>need-driven</u> in the sense that work at each stage, whether by the speaker or one of the transducers, was postponed as long as possible; and when the need to generate further words forced one or more components to do some work, they did the minimum necessary. This was because, for reasons of efficiency and psychological plausibility, McDonald wanted to avoid backtracking at any stage. If any stage of the realisation process for one element might impose a constraint on some stage of the realisation of a later one (for example, if they represented an initial and subsequent reference to the same entity), it was desirable for the earlier element to be completely processed before processing started on the later one. We will see in section 9.4 that SPAR's generator is need-driven in the same way, and for similar reasons.

McDonald's MUMBLE system, which was essentially an implementation of the linguistic component of this model, generated impressive output from the messages supplied to it by a number (six, in McDonald [1981]) of notional "speakers" for different domains and representations. This suggests that the model is also likely to be applicable to SPAR's requirements.

The other three generators we will look at, selected because of their particular relevance to SPAR, are all essentially tactical ones; that is, they are analogous to the linguistic component in McDonald's model rather than to the whole generator. The messages they are to convey, and the order in which to convey them, are given by other systems or system components rather than needing to be calculated dynamically.

(2) <u>Herskovits</u>' generator derived French output from the template-based semantic blocks with which Wilks' PS system represented texts. Each word sense formula and (preposition-defining) paraplate in PS's English dictionary had attached to it one or more alternative "stereotypes": context-sensitive patterns which specified how to express that word sense in French. The generator treated the semantic block as a transition network, traversing it and evaluating the stereotypes it encountered. Evaluating a stereotype might involve recursively evaluating other, dependent ones. The result would be either a French string, which would be added to the text being constructed, or a failure, which would cause the generator to back up and try an alternative

stereotype. Thus Herskovits' generator differed from McDonald's linguistic component in two important ways: it was non-deterministic, and (like the analyser at the other end of PS) it made no use of an intermediate syntactic representation.

(3) Boguraev's generator was designed in accordance with his view of paraphrase summarised in 9.1.1 above, and performed operations such as synonym selection and constituent reshuffling. It needed no strategic component because the appropriate answer to the question "what to say" was always the trivial one: "all of, and only, the dependency structure in question". Just as McDonald's linguistic component contained two transducers linked by an intermediate surface structure representation, so Boguraev's generator constructed a sentence in two stages. The first stage selected a contextually appropriate verb to express the main verb sense of the dependency structure, and using semantic and syntactic information associated with that verb constructed a linearly ordered, hierarchical, syntactico-semantic *environment network*. This network contained semantic case primitives such as AGENT and DESTINATION, syntactic categories such as PREP-PHRASE and NOUN, and English words and phrases. Boguraev defended this heterogeneous representation by arguing that semantic as well as syntactic information was necessary for certain aspects of the second stage of generation: that of traversing the environment network left-to-right, outputting the words in it after operations such as subject deletion in some embedded clauses and verb and noun morphology.

(4) Tait's generator, which also took Boguraev's dependency structures as input, was developed as part of a document retrieval project to facilitate the generation of alternative text forms (either whole sentences or constituents) for underlying dependency structure representations of concepts. Unlike Boguraev's generator, which produced only one sentence per dependency structure, Tait's generates as many alternative paraphrases per structure as possible in order to maximise the chance of the retrieval search succeeding; however, it is also capable of operating in single-paraphrase mode.

Tait's generator, like Boguraev's and like McDonald's linguistic component, is a two-stage one; the intermediate representation is a broad, flat surface syntactic phrase structure tree in the style of Winograd [1972]. Both stages of generation are data-driven, depth-first and left-to-right; however, the first stage is completed before the second is begun, so, like Boguraev's, the generator is not need-driven (and indeed there was no reason for either generator to be so).

An important feature of Herskovits', Boguraev's and Tait's generators, and indeed of SPAR's, was their use of alternative, context-sensitive rules to express a given word sense. The words and constructions generated were not derived by a global examination of the sentence representation, as they were in, say, Goldman's [1975] BABEL (the generator for the CD-based MARGIE system); instead, output patterns were associated in the dictionary with input word senses and preserved throughout processing. Boguraev stresses that driving his generator by means of word senses is purely a matter of convenience; it would still be possible, much as in BABEL, to arrive at an appropriate choice of words if only formulas (and not word senses themselves) were present in dependency structures. However, this is emphatically not the case in SPAR, because the use of the THIS primitive to indicate missing information (see 4.2.1) means that two very different word senses can have the same formula: e.g. both ELEPHANT1 and SHEEP1 are defined as (THIS BEAST). Therefore, in accordance with the lack of any important distinction between primitives and word senses in SPAR's representational formalism, SPAR's generator is unavoidably word sense based in a way that BABEL, and in principle Boguraev's generator, were not.

9.2.2 The components of SPAR's generator

We have seen that SPAR's generator must produce a paraphrase of each sentence which makes all resolution decisions as clear as possible. In order to take advantage of contextual information (i.e. information derived from earlier sentences), the generator is invoked after a sentence has been fully processed; that is, after all its anaphors have been resolved and the current fragment has been merged into the context. A paraphrase of the sentence can then be produced by expressing in English the event represented by the TMN node corresponding to the main clause of the sentence, which will of course have subsidiary clause information (if any) associated with it.

SPAR's generator has a strategic and a tactical component. The strategic component decides what to say, i.e. what part of the information in the TMN to express, and packages that information as a message using a suitable representation; the tactical component then converts the message into English.

The representation used for such messages is that of Boguraev's dependency structures. This representation was selected because:

(1) It is a tried and tested one, and has been successfully used for a number of applications, including paraphrase generation.

(2) SPAR's TMN was originally derived from Boguraev's structures, to which, as we saw in chapter 4, it has many similarities, so the inverse transformation should be comparatively straightforward.

(3) If the strategic component of the generator were to output Boguraev-style messages, considerable effort could be saved by using either Boguraev's or Tait's generator (both of which were available) as the basis of the tactical component.

Tait's generator, rather than Boguraev's, was selected for the tactical component for the following theoretical and practical reasons.

(1) Its intermediate representation (straightforward phrase structure trees) is rather more conventional and well-defined than Boguraev's mixed syntactic and semantic environment network. In particular, certain syntactic transformations which we will see later are desirable are more easily expressed in terms of phrase structure than in terms of environment networks.

(2) As Sparck Jones and Tait point out, Boguraev's generator accepted the original [1979] version of his representation language; since then the representation has undergone a number of changes. Tait's generator, however, accepts the more recent version of the representation now produced by the analyser and accepted by SPAR.

(3) The data-driven style in which Tait's generator is written makes it easy to understand and modify.

The components of SPAR's generator, and the representations they accept as input and produce as output, are therefore essentially as shown in figure 9.2. The approximately corresponding parts of McDonald's framework are given for comparison.

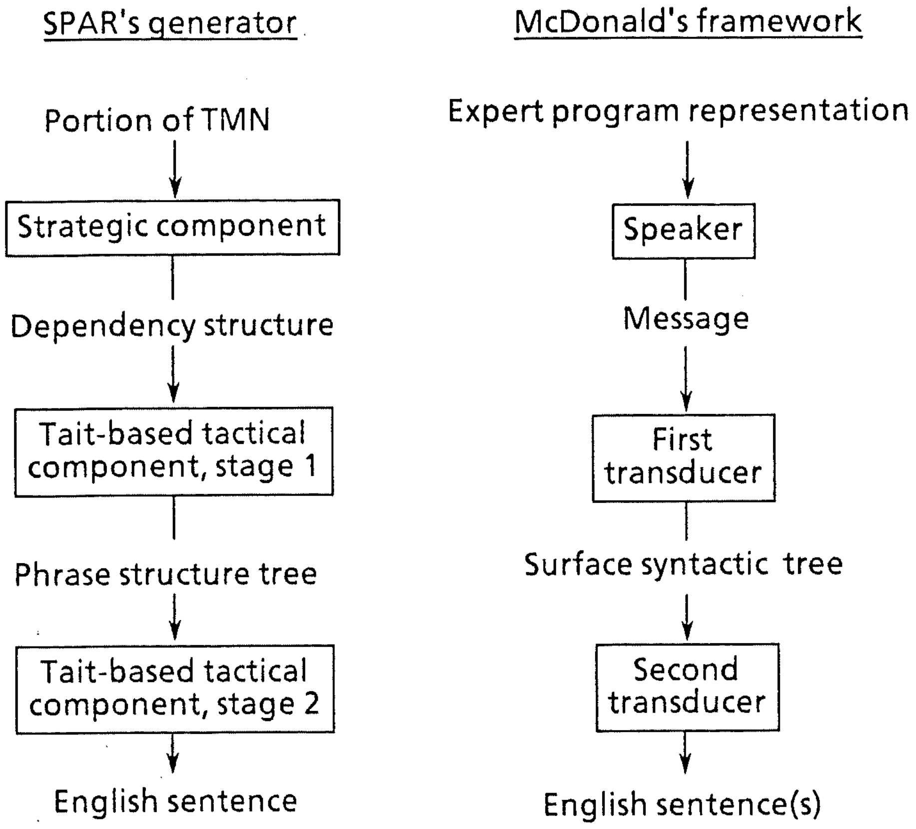


Figure 9.2: SPAR's generator compared with McDonald's framework

In the remainder of this chapter, we will fill out the details of this figure, and in particular show how and why the whole process is controlled in a need-driven way.

9.3 Some guidelines for generating suitable paraphrases

So far in this chapter, we have set out the general goals we are trying to achieve with our paraphrases and the general structure of the generator which will construct those paraphrases. From now on, we will be more specific. In this section, we take the problem of defining the kinds of paraphrases we want, and establish some specific linguistic guidelines or heuristics for ensuring, as far as possible, that the interpretations assigned to our three kinds of ambiguity (word-sense, structural and anaphoric) are clearly expressed. These guidelines will be specified in terms of the three successive representations used by the generator (the initial TMN, Boguraev's dependency structures and subsequently Tait's surface syntactic phrase structures).

The task of expressing clearly the interpretations of anaphoric ambiguities falls mainly to the strategic component of the generator, which must ensure that the messages it hands to the tactical component are sufficiently informative. The paraphrase of an anaphor must describe the anaphor's specification unambiguously: that is, it must make clear what the anaphor's antecedent is and how it relates to that antecedent. For example, for the text

(9-1) [1] John sold his old car
[2] He bought a new one.
[3] It broke down within a week.

the paraphrase "The car broke down within a week" for [3] would not show what specification had been chosen for "it"; to avoid ambiguity, we would need to say that it was "the new car" or "the car that John bought" that broke down.

On the other hand, the task of expressing the resolutions of lexical and structural ambiguities falls mainly to the tactical component of the generator, which must word the information handed to it so as to avoid, if possible, the ambiguities in the original input sentence. This will involve actions such as selecting synonyms and moving constituents.

Because the strategic aspect of generation is, at least at the level of sophistication of SPAR's generator,[1] logically prior to the tactical one, we will first consider what the strategic component must do to show the results of resolving anaphoric ambiguity. We will then look at what the tactical component must do to avoid the lexical and structural ambiguities in the input.

9.3.1 The strategic component: avoiding anaphoric ambiguity

As we have seen, the strategic component of SPAR's generator takes as input the TMN node derived from the main clause of the current sentence, and gives as output a dependency structure which it hands to the tactical component. To construct such a dependency structure, the strategic component must therefore

(1) decide what set of TMN nodes and links (**Corresponds:** assertions) should be exploited, and

(2) turn those nodes and links into a dependency structure.

Because of the similarities between the TMN representation and Boguraev's, subtask (2) is not very difficult, and will not be discussed in detail. Essentially all that is required is the inverse of the process of constructing a "skeleton fragment" from a dependency structure, as described earlier in 4.3.3.

---

[1]Danlos [1984] shows that imposing a fixed order on decisions (e.g. conceptual before lexical) may remove some or even all of the options available for later decisions, and that therefore flexible control structure, based on interactions between decisions, is needed. To overcome such problems were they to arise, SPAR's generator could perhaps be made non-deterministic: if an earlier decision resulted in no realisation being possible, an alternative could be tried.

The substantive part of the strategic component's task is therefore subtask (1): deciding what set of TMN nodes and links to exploit, or, in more abstract terms, deciding what objects, events and actions to mention. Such decisions are crucial if the paraphrase is to convey all the conceptual content of the input with as little redundancy as is consistent with making resolution decisions clear, and in particular if its component noun phrases are to be referentially adequate (in the strong sense given in 9.1.2) but still as efficient as possible.

Subtask (1) is a recursive one. Given a TMN node, the generator must decide what neighbouring nodes must also be included as a direct consequence; and the same decision must be made for each of these further nodes. Thus the following decisions should be made when, after anaphor resolution, the generator is handed the TMN node for the main clause of [2] in our example (repeated from chapter 4)

(9-2) [1] John was angry.
[2] He wanted to break a window.

The generator should decide that the TMN nodes representing both the agent of wanting (John) and the object (the breaking action) should also be expressed. Expressing the latter will in turn involve expressing its agent and object (although the agent, which is also the "John" node, may in fact eventually be realised as the null string).

Thus to accomplish subtask (1), we need to formulate rules for deciding what extra TMN nodes must be included to express adequately a given TMN node; these rules will then be recursively applied to the extra nodes. We will now set out the rules SPAR's generator uses, first those for nodes that represent non-anaphors (or anaphors that SPAR has failed to resolve) and then for those that represent resolved anaphors.

When a node not representing a resolved anaphor is encountered, the generator mostly aims to mirror the organisation of the input dependency structure; this is possible because of the close resemblance between input structures and the TMN, and in particular because of the existence of the E ("essential") flags on `Corresponds:` assertions (see 4.3.2). Thus only explicit nodes (i.e. nodes derived directly from input phrases) connected to the current one by "essential" `Corresponds:` assertions, and therefore representing restrictive modifiers or dependents, will be included. This ensures that, except where anaphors are concerned (see below), each phrase in the the input sentence corresponds to something in the paraphrase and vice versa, thus making comparison and evaluation easier (but not, as we will see, preventing the construction of interesting paraphrases).

In order to clarify certain ambiguity resolutions, non-restrictive modifiers, which in fact are always relative clauses, are conjoined to the main clause in the paraphrase; thus "I stretched out my hand to the monkey who bit it" would receive a paraphrase such as "I stretched out my hand to the monkey and the monkey bit my hand" if (and only if) SPAR had concluded that the relative clause was non-restrictive.

Noun phrases not resolved as anaphors of any kind are always given an indefinite article ("a(n)" or "some") even if the original noun phrase was definite. Thus if a story begins "John went to the zoo", the paraphrase will be "John went to a zoo" because (not surprisingly) no pre-existing zoo has been found in the TMN. Although "a zoo" is arguably not an exact paraphrase of "the zoo" here, the use of definite articles only for resolved anaphors means that the reader of a paraphrase can tell at a glance whether or not an input noun phrase has in fact been resolved. (I emphasise this point to avoid misunderstanding of SPAR's output).

We will now look at the task of paraphrasing resolved anaphors, beginning with the case of an anaphor which has been recognised as specifying an explicit contextual node (i.e. a node derived directly from an earlier input phrase). Since such a node will already have been mentioned in an earlier paraphrase, what we need to do is to include enough information to distinguish that node from all other explicit ones. For sentence [2] of (9-2), "he" (=John) can be expressed by "John" alone, while in our earlier example (9-1), the "it" in sentence [3] can be expressed by phrases like "the new car" or "the car that John bought" to distinguish it from the old car mentioned in [1].

This degree of explicitness is not always necessary to ensure that the paraphrase is unambiguous; for (9-2), the "he" in [2] could unambiguously be "paraphrased" by "he", since only one male human has been mentioned. However, using "he" would hardly be convincing evidence that SPAR had resolved the pronoun. Similarly, in (9-1), the paraphrase "the car" for "it" would be unambiguous to the reader because the new car is more focussed, but again such a paraphrase would not reflect SPAR's selection of antecedent. It is therefore necessary to incorporate at least enough information to distinguish the specification from all other explicit nodes; but a pronoun should not be used, even if adequate in this respect, because it provides no direct evidence for our demonstration purposes that the original anaphor has been resolved at all.

A consequence of not using pronouns is that paraphrases will often break the c-command rule and therefore might be thought ungrammatical or misleading. If we paraphrase

(9-3a) John promised Bill that he would arrive soon.

by

(9-3b) John promised Bill that John would arrive soon.

then the second "John" in (9-3b) is c-commanded by the first and, according to Reinhart's original [1976] theory, cannot corefer with it. This objection can be countered by appealing to Reinhart's more recent [1983] theory, in which c-commanding implies non-coreference only if, in the hearer's view, the speaker has no reasons to avoid using bound-variable (=pronominal) anaphora. But, as we have seen, SPAR's generator does have such reasons, which must be kept in mind when reading the paraphrases.

We saw above that when paraphrasing sentence [3] of (9-1), a modifier such as "new" or "that John bought" was necessary to identify the correct car. This raises the question of how modifiers should be selected from among those available. In order to find (an approximation to) the shortest possible modifier, SPAR examines modifiers likely to be realizable as adjectives, prepositional phrases, simple relative clauses and complex relative clauses,[2] in that order, and accepts the first one that distinguishes the node in question from all others. Thus whereas in (9-1), the adjectival modifier "new" distinguishes one car from the other, in a text like

(9-4) [1] Bill bought a new car.
[2] John bought a new car too.
[3] It broke down within a week.

the modifier "new" would be insufficient, and something like "that John bought" would be used instead. (Although the possessive relationship between John and the car is not explicitly present in the TMN, the generator could in principle be extended to infer it, thus allowing the modifier "John's". However, such inference might sometimes detract from referential adequacy; it is clearer to identify the car by means of a relationship that has already been explicitly stated).

However, if the node in question has already been paraphrased in the current sentence, no modifiers are sought; a simple noun phrase of the form "that X" or "those Xs" is virtually always adequate to identify the correct specification.[3] The use of simple noun phrases on second and subsequent occasions of mention in a sentence is essential if paraphrases are not to become too repetitive. Evidence for this will be given at the end of the chapter.

Having looked at the treatment of non-anaphors and of anaphors directly cospecifying with their antecedents, we now consider anaphors related to their antecedents indirectly, by associated or inferred specification. In such cases, it is desirable to state explicitly the relationship between anaphor and antecedent. For example, in

(9-5) [1] The heiress lived as a recluse.
[2] She died under mysterious circumstances.
[3] The murderer was not found.

we want to generate a phrase like "the murderer who killed the heiress", or, to avoid tautology, "the person who killed the heiress". (A briefer description

---

[2]By a "simple" relative clause I mean one like

the car that John bought.

where the head noun phrase is a direct case filler of the main verb of the relative clause. A "complex" relative clause would be one like

the car that John wishes Bill had not persuaded him to buy

where "the car" is a case filler not of "wishes" but of "buy". (In fact the generator cannot at present properly express complex relative clauses, as example B9 in appendix B shows).

[3]The determiners "that" and "those" are preferable to "the" because they do not allow implicit cospecification. That is, whereas "a zoo...the monkeys" is acceptable, "a zoo...those monkeys" is awkward unless the monkeys have already been explicitly mentioned. The use of "that" or "those" therefore restricts the interpretation options and minimises the risk of ambiguity in the output.

such as "the heiress's murderer" is less referentially adequate because it does not make clear the exact nature of the ambiguity resolution; for example, it would be consistent with SPAR having erroneously decided that the murderer was employed by the heiress as a hired killer).

When faced with a node representing an indirect anaphor such as "the murderer" above, SPAR looks for the shortest route in the TMN to an already-paraphrased node, and makes that route into a modifier. In (9-5), when a paraphrase for "the murderer" is sought, the shortest such route is that via the implicit "kill" node (derived from the "murderer" formula) to the already-paraphrased "heiress" node. Having found such a modifier, SPAR finds that, according to information in the WSN, the "kill" modifier exhausts the distinction between MURDERER1 and the immediately dominating MAN node in the WSN Specialisation: hierarchy. It therefore inserts MAN (which is eventually expressed as "person") rather than MURDERER1 into the dependency structure to avoid redundancy.

9.3.2 The tactical component: avoiding lexical and structural ambiguity

Once the strategic component of the generator has built a dependency structure, it is the job of the tactical component to express it in English as clearly and unambiguously as possible. Like Boguraev's generator, it does this by choosing suitable synonyms for ambiguous words (because of lexical ambiguity) and by ordering and transforming constituents (because of structural ambiguity).

SPAR's synonym selection is fairly rudimentary. Each word sense has associated with it, in a synonym dictionary, zero or more roughly synonymous senses. A number of case name transformations may be associated with a synonym; for example, using the synonym SELL1 for BUY1 entails transforming the AGENT to the RECIPIENT and the ABSTRACT-SOURCE to the AGENT. Only if the strategic component has provided both an AGENT and an ABSTRACT-SOURCE will the synonym be acceptable. The first acceptable synonym is used; if there are none, no substitution is made. (Of course, the word selected as a synonym may on its own be just as potentially ambiguous as the original; however, this ambiguity should disappear if the paraphrase is read with the original in mind).

Each word sense covered by the analyser also has an an entry in the dictionary used by Tait's generator (or if it does not, SPAR constructs one dynamically, and usually fairly reliably, using analyser dictionary information). An entry specifies one or more ways in which a structure headed by that word sense may be realised, giving details of syntactic category, root form, and morphological and syntactic irregularities. Thus SPAR, like Herskovits' and Boguraev's generators, embodies a "fanning out" mechanism where each word has several senses and each word sense may be expressed in several ways. As we observed earlier, word senses are essential to generation because the THIS mechanism prevents senses, and therefore appropriate synonyms, being deduced from formulas alone, as they could in principle have been in Boguraev's system.

We now turn our attention from lexical to structural ambiguity. Virtually all of the structural ambiguities in the stories SPAR has processed concern prepositional phrase (PP) attachment. Structures likely to be realised as PPs are therefore subjected to two types of transformation.

The first type is applied to certain cases attached to noun-args (representing nominals) in the dependency structure; it results in relative clauses rather than prepositional phrases being generated. The noun-arg

```
(N
   (MAN1 MAN
      (@@ LOCATION PARK1-1) 4
      (@@ DET (THE1 ONE))))
```

which would be realised as "the man in the park", is transformed to

```
((TRACE (CLAUSE V AGENT))
 (CLAUSE
    (TYPE REL)
    (TNS PAST)
    (V
       (BE1 BE
           (@@ AGENT (N (MAN1 MAN (@@ DET (THE1 ONE)))))
           (@@ LOCATION PARK1-1)))))
```

which is realised as "the man who was in the park". Thus if SPAR decides that in a certain story context, the structurally ambiguous sentence "I saw the man in the park", "in the park" attaches to "man" rather than "park", the paraphrase produced is "I saw the man who was in the park", in which the ambiguity is not present.

The second type of transformation is applied at a later stage, when the dependency structure has been turned into a surface syntactic tree. At this stage, in order to avoid potential attachment ambiguity in the output paraphrase, a PP structure in a finite clause is preposed if it is preceded either by a noun phrase or by another PP structure. The transformation effects changes such as these:

I saw the man in the park → In the park, I saw the man.

I rode towards the man on the horse →
On the horse, I rode towards the man.

Together, the two transformations (PP relativisation and preposing) separate out the five readings delivered by the analyser for the sentence "I saw the man in with park with the telescope". Like Boguraev's generator, the tactical component of SPAR's generator, when applied to each of the five structures built by the analyser,[5] gives:

---

[4]The underlined item PARK1-1 is a TMN node which has not at this stage been converted to a noun-arg of the form (N (PARK1 ...)). This is because of the adoption of the incremental, need-driven approach which will be discussed fully in the next section.

[5]The tactical component of the generator can, for testing and demonstration purposes, accept input directly from the analyser as well as from the strategic component. It is able to deal with embedded TMN nodes, but does not demand their presence.

```
I SAW THE MAN WHO WAS IN THE PARK WHICH HAD THE TELESCOPE.
I SAW THE MAN WHO WAS IN THE PARK AND WHO HAD THE TELESCOPE.
WITH THE TELESCOPE, I SAW THE MAN WHO WAS IN THE PARK.
IN THE PARK WHICH HAD THE TELESCOPE, I SAW THE MAN.
IN THE PARK, AND WITH THE TELESCOPE, I SAW THE MAN.
```

although of course only one of these will be generated in any given context, depending on the context-determined choice SPAR has made.

9.4 The detailed organisation of the generator

In this section, we will conclude the detailed discussion of the generator by showing why a need-driven approach is necessary and, by means of a simple example, how it is implemented. We will then look in detail at a more complex example which brings together many of the points we have covered in the chapter.

9.4.1 A "need-driven" control structure

So far, we have seen that SPAR's generator has a strategic and a tactical component. The strategic component builds a dependency structure from a TMN node and its environment in the TMN; it is responsible for ensuring that enough information is included to make clear the specifications of anaphors. The tactical component is an augmented version of Tait's two-stage English generator, which turns a dependency structure into an English sentence by way of a surface syntactic phrase structure. SPAR extends Tait's generator partly by improving its treatment of morphology and its range of syntactic constructs, but more importantly by adding components that transform the intermediate dependency and phrase structures in order to clarify the resolution of lexical and structural ambiguity. The full set of processing steps is as shown in figure 9.3.

Although figure 9.3 applies locally, in the sense that every piece of information expressed will pass through all the stages in the order given, closer examination of the manipulations performed at each stage reveals that the process cannot be a simply linear one; that is, we cannot just apply each process in turn to all the information involved. We saw in 9.3.1 that when the strategic component is generating a piece of dependency structure from a nominal TMN node, it must take account of whether that node has been mentioned earlier in the current sentence; if so, a structure which will eventually be realised as "that X" will be sufficient, and indeed the detailed description necessary for the earlier mention (e.g. "the X which John...") will be far too verbose. However, if the generation process is purely linear, the strategic component will not know which occurrence of a TMN node will be realised first, because word order is largely decided by the tactical component which will not yet have been invoked.

To solve this problem, a more sophisticated, need-driven control structure is adopted, in which the components of the generator communicate with one another in both directions. This control structure resembles that proposed by McDonald.

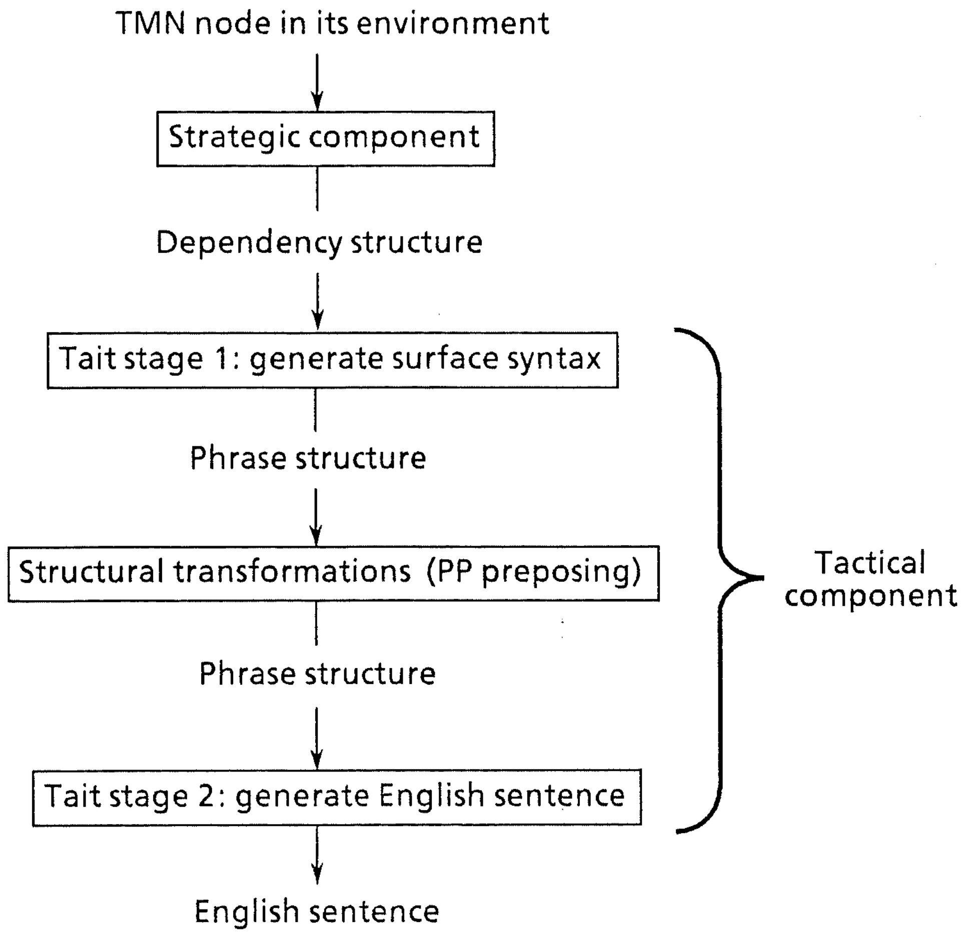


Figure 9.3: Stages in SPAR's generation process

When the strategic component is handed a TMN node, it produces a piece of dependency structure for that node, but does not recursively call itself on neighbouring nodes. Thus when SPAR processes a story beginning "John went to the zoo", it constructs a TMN node G01-1 to represent the "went" action. When the sentence has been fully processed by SPAR, this node is handed to the generator's strategic component, which, rather than passing a complete dependency structure on to the tactical component, passes on the hybrid structure

```
(CLAUSE
   (TYPE DCL)
   (TNS PAST)
   (V
      (GO1 MOVE
         (@@ AGENT JOHN1-1)
         (@@ DESTINATION ZOO1-1))))
```

where the underlined objects are TMN nodes. These TMN nodes are carried forward in the tactical component until the word order of (this level of) the sentence has been finally decided, i.e. until after any PP-preposing transformations (there are none in this case) have been applied to the phrase structure. The second stage of the tactical component, which walks over a phrase structure depth-first and left-to-right to derive an English string, reinvokes the whole generator recursively on any TMN node it comes across, expecting it to return an English string. In our example, the phrase structure (in a simplified form) is

```
(INTRANSITIVE-CLAUSE-PATTERN
  JOHN1-1
  "WENT"
  (PREP-PHRASE "TO" ZOO1-1))
```

When the second stage of the tactical component is applied to this structure, it first causes the generator to be applied recursively to the TMN node JOHN1-1; this ultimately produces (via simple dependency and surface structures) the single-word string "JOHN", which is output, and is followed by the words "WENT" and "TO", taken directly from the top-level phrase structure. The generator is then invoked again, this time on the TMN node ZOO1-1; the string "A ZOO", is returned and output. Thus the generation of "JOHN" is completed before the generation of "A ZOO" is begun.

In fact the whole sentence generation process is almost entirely left-to-right. For any two clauses or noun phrases in a sentence, the phrase to the left is fully generated before the generator is first applied to the one to the right except in the case that one of the phrases includes the other. In particular, if a sentence contains multiple references to the same entity, each reference will have been fully generated by the time the next is begun. The generator keeps a record of what TMN nodes have been expressed so far in the current sentence; this allows it to make second and subsequent references of the form "that X" or "those Xs" in order to satisfy the efficiency requirement.[6]

9.4.2 A detailed example

To illustrate the generation process as a whole, and the need for a need-driven approach in particular, we will now look at the process of generating paraphrases for the second sentence of the text

(9-6) [1] John went to the zoo.
[2] He saw the elephants waving their trunks.

The paraphrase SPAR produces for this text, the first sentence of which we have already seen, is

---

[6]Indeed, because the "expressed so far" record is updated just before the generator starts work on a node, rather than just after it finishes, it can deal correctly with the case when a noun phrase cospecifies with another embedded in it. For example, "a country which exports petrol to its neighbours" could, under the relevant interpretation, be paraphrased correctly as "a country which exports petrol to <u>that country's</u> neighbours" because the generation of the outer reference (the entire NP) would have ***begun*** (though obviously not ended) before that of the inner (possessive) one; the TMN node would therefore be flagged "expressed" by the time the inner reference was begun. This process would only fail to give appropriate results if the embedded reference occurred ***before*** the matrix (head) NP, which seldom if ever occurs in English.

(9-7) [1'] John went to a zoo.
[2'] While the elephants which were among the animals which were at the zoo waved those elephants' probosces, John saw those elephants.

Paraphrase [2'] is necessarily long-winded because it has to express five ambiguity resolutions: the decisions that

(1) "he" is John;

(2) "the elephants" are the ones associated with the zoo;

(3) "waving their trunks" is not a restrictive modifier; i.e. it tells us what the elephants were doing when John saw them, rather than distinguishing between some elephants who were waving their trunks and some who weren't;

(4) "their" is the elephants;

(5) "trunks" is used in the sense of probosces, and not luggage or swimming trunks.

An important point to note about [2'] in the output paraphrase is that, because of the need-driven nature of the generation process, the elephants are only described in full the first time they are mentioned; on the two subsequent occasions they are referred to as "those elephants". This is necessary to avoid the excessive repetitiveness and opacity of

[2''] ?? While the elephants which were among the animals which were at the zoo waved the probosces of the elephants which were among the animals which were at the zoo, John saw the elephants which were among the animals which were at the zoo.

A need-driven approach is required because the strategic component of the generator must provide a full description of the elephants only on the first occasion that they are mentioned, and not on the second and third occasions. For this to be possible, the tactical component must already have decided much of the word order of the sentence, and in particular, which mention of the elephants is to come first.

The production of the paraphrase [2'] occurs as follows. The strategic component derives from the TMN node SEE1-1, which represents the "seeing" event described by [2], the dependency structure

```
(CLAUSE
   (TYPE DCL)
   (TNS PAST)
   (V (SEE1 SENSE
         (@@ AGENT JOHN1-1)
         (@@ OBJECT ELEPHANT1-1)
         (@@ TIME-SPAN WAVE2-1))))
```

Next, the surface structure

```
(TRANSITIVE-CLAUSE-PATTERN
   JOHN1-1
   "SAW"
   ELEPHANT1-1
   (PREP-PHRASE-PATTERN "WHILE" WAVE2-1))
```

is derived by the first stage of the tactical component. To avoid possible structural ambiguity, the prepositional phrase is then preposed to give

```
(PREPOSED-PATTERN
   (PREP-PHRASE-PATTERN "WHILE" WAVE2-1)
   (TRANSITIVE-CLAUSE-PATTERN
      JOHN1-1
      "SAW"
      ELEPHANT1-1)
```

This is handed to the second tactical stage, which attempts to read off an English string from it. The word "WHILE" is output, and then the generator is invoked recursively on the TMN node WAVE2-1. This produces a dependency structure which is transformed to the surface structure

```
(TRANSITIVE-CLAUSE-PATTERN
   ELEPHANT1-1
   "WAVED"
   TRUNK1-1)
```

The attempt to read off an English string from this triggers yet another invocation of the generator, this time on the node ELEPHANT1-1. The strategic component discovers that this is the first time the elephants have been mentioned, and attempts to describe them in terms of some previously-described object or event. It discovers that they are quite closely related to the zoo, and produces a large dependency structure which is eventually realised as "THE ELEPHANTS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO". This string is output, followed (from the next level up) by "WAVED".

The generator now tries to express TRUNK1-1, which in turn (the strategic component decides) involves mentioning the elephants. This time, because they have already been mentioned in the current sentence, the strategic component produces a much smaller dependency structure which is eventually realised as "THOSE ELEPHANTS". The synonym PROBOSCIS1 is selected for the sense TRUNK1 to show that the word sense ambiguity (between elephants' trunks, luggage and swimming trunks) has been resolved.

On the third occasion that the elephants are mentioned, after the words "JOHN SAW" are output, the words "THOSE ELEPHANTS" are once more produced, again because of the earlier mention. The final paraphrase, repeated here with the references to the elephants underlined for comparison, is therefore

> [2'] While <u>the elephants which were among the animals which were at the zoo</u> waved <u>those elephants'</u> probosces, John saw <u>those elephants</u>.

9.5 Summary and evaluation

At the beginning of this chapter, we posed two questions for which answers had to be found in order to provide SPAR with a generator which would enable its performance in ambiguity resolution to be evaluated. The questions were:

(1) What sort of paraphrases should the generator produce? and

(2) How can the generator derive such paraphrases from SPAR's meaning representation?

Our answers to these questions, first in general and then in specific terms, were as follows.

To deal with (1), we examined the arguments Boguraev [1979] used in designing a paraphrase generator for his analyser. We found we had to adapt the requirements that Boguraev placed on his generator, and the kinds of operations he believed were necessary to satisfy those requirements, to the case of SPAR. Thus we concluded that the task of SPAR's generator was:

> to produce, for each input sentence in the story, an output sentence which is intuitively acceptable in the context as a paraphrase, and which shows what decisions SPAR has made in resolving ambiguities, especially anaphoric ones.

To deal with question (2), we exploited McDonald's comprehensive need-driven model of the generation process, and Tait's two-stage tactical generator for Boguraev's dependency structures. Thus using McDonald's framework, we decided that SPAR's generator should have a strategic component which derived a dependency structure from a region of the TMN, and a tactical component, based on Tait's generator, which turned the dependency structure into an English sentence.

More specifically, for question (1), we developed a set of guidelines for the strategic component to follow to ensure as far as possible that each paraphrase would contain enough information to avoid any possible ambiguity, but not so much as to be tautologous or absurdly repetitive. For the tactical component, we outlined some procedures for synonym substitution and structural transformation which would help to remove lexical and structural ambiguity respectively.

For question (2) in turn, we asked ourselves how the generator could carry out the operations we had just decided were necessary. We saw that we required a need-driven control structure, in which work was postponed until the need to output a word demanded it. Under such a control structure, earlier parts of a paraphrase are fully generated before later ones are begun.

Like the other parts of the system, SPAR's generator is best assessed on its performance. To facilitate this, the results it produces for a selection of texts are shown in appendices A and B. These results suggests that the generator achieves its goal of producing paraphrases which show what ambiguity resolutions have been made. The paraphrases are, by and large, not easy to read on their own, but ease of reading was not one of our requirements; we merely required them to be grammatical and comprehensible.

The originality of the work discussed in this chapter lies mainly not in the individual ideas discussed, many of which are drawn from others' work (in particular, that of McDonald, Boguraev and Tait), but in the way those ideas are combined to create a "full" (i.e. both strategic and tactical) generator which differs in its aims from most of those written to date. Most language processing applications involving the production of text require that that text normally be as easy to read and "user-friendly" as possible, but for SPAR, ease of reading is secondary to elimination of ambiguity. The techniques developed here therefore may not be applicable to what most applications need most of the time. However, there are many applications for which paraphrases of the type SPAR produces would sometimes be very useful. For example, if an interactive machine translation or database query system is unable to resolve some ambiguity in its input, it may be desirable to present the options to the user in natural language form and ask for the correct one to be indicated. A generator such as the one presented here would be able to make the differences between the various options clear, whether those differences were caused by lexical, structural or anaphoric ambiguity.

10. Conclusions

> "Life is the art of drawing sufficient conclusions from insufficient premises."
>
> *Samuel Butler, Notebooks, "Life", 9*

In previous chapters, we looked at some of the problems of anaphor resolution, and saw in detail how SPAR tackles those problems in a shallow processing framework based on the theories of Wilks and Sidner. This last chapter takes a more global view. First, the goals of the work are restated, and the SPAR system is summarised as an attempt to achieve those goals. Next, the major components of the system are assessed, not so much in terms of their intrinsic theoretical content, which was evaluated at the ends of earlier chapters, but rather in terms of their contributions to the overall system performance and also in terms of more practical considerations such as simplicity, reliability and extensibility. This assessment leads on naturally to a more general discussion of the viability of doing anaphor resolution by shallow processing. Finally, the possibilities for improving and extending the system in various ways are discussed.

10.1 System review

This thesis began with a statement of four interrelated goals.

The first goal was to develop a system which would resolve ambiguities, especially anaphoric ones, in simple stories, and generate paraphrases to show what resolution decisions had been made.

The exploitation of world knowledge by inferential processing has a place in achieving this first goal. However, systems capable of powerful, flexible and reliable inference about non-specialised domains are still beyond the state of the art. The second goal, therefore, was to investigate the possibility that the first goal might instead be achieved by a shallow processing approach. In such an approach, linguistic knowledge is used as much as possible; the inference component is invoked as little as possible, and has access only to limited quantities of explicit world knowledge.

Wilks' theory of preference semantics was selected as a suitable basis for the system's inference component and word-sense-level semantics. Because a system that resolves anaphors must be able to keep track of focus, especially local focus, Sidner's theory of local focussing was also introduced. The third goal of this work was to adapt, test and develop Wilks' and Sidner's theories, both of which seemed very promising but somewhat undertested.

The fourth goal was to find a way of coordinating the focussing, inference, and other components of the system so as to maximise accuracy while minimising the amount and complexity of processing. Such coordination is crucially important if the inevitable failings of inference in a shallow processing framework are to not to mislead the system as a whole into making wrong decisions.

Together, these four goals constitute an attempt to substantiate the major theoretical claim of the thesis: a particular form of "shallow processing hypothesis", to the effect that

> a story processing system which exploits linguistic knowledge, particularly knowledge about focussing, as heavily as possible, and has access only to limited quantities of world knowledge, which it invokes only when absolutely necessary, can usually choose an appropriate antecedent for an anaphor even in cases where the inference mechanism by itself cannot do so.

SPAR represents an attempt to achieve the goals stated above and to test the shallow processing hypothesis. It accepts input from Boguraev's analyser, converts it to a text model form based largely on a developed version of Wilksian word-sense semantics, and resolves anaphors using a Sidnerian framework significantly extended to deal with intrasentential antecedents, multiple anaphors in the same sentence, and indefinite as well as definite anaphors.

Components exploiting syntax, semantics, local focussing and world knowledge are all coordinated in the effort to resolve anaphors. Their predictions are reconciled by an arbitration mechanism which recognises that some knowledge sources are more reliable than others. Such a mechanism is essential in a shallow processing system where the results of inference will often be incomplete and/or unreliable.

The inference component is based on Wilks' ideas but goes significantly beyond them in several ways, for example to include object-oriented and goal-oriented reasoning. It is applied only when necessary, and then as late as possible. If the arbitrator cannot reach a firm decision even after taking inference into account, some further "collective" heuristics are applied; these heuristics guide the system to the right resolutions in most of the texts processed.

If Boguraev's analyser is unable to resolve all the word-sense and structural ambiguities in a sentence, it produces several alternative interpretations. When this happens, SPAR processes each interpretation as just described. A score is calculated for each interpretation, based on its semantic density (as determined by detailed comparison with the relevant word sense definitions) and the ease with which anaphors can be resolved. The highest-scoring reading is accepted.

When all the ambiguities in a sentence have been resolved, it is merged into the context and a paraphrase of it is produced which shows what resolution decisions have been made. The generator which produces the paraphrase makes both strategic and tactical decisions; it operates on a need-driven, left-to-right basis to ensure that appropriate descriptions are generated at each point in the sentence.

10.2 Assessment of the system components

This technical assessment of the major components of SPAR has a practical emphasis, and is based on the experience of implementing, developing and testing the ideas reported. Nevertheless, it also has theoretical implications, because computational theories of (a given aspect of) language processing are to be judged largely by the ease with which they can be implemented and made to contribute to the reliable performance a reasonably complete language processing system.

We will therefore concentrate on assessing the contribution made by each component to the overall system performance. For example, we will not ask whether the inference component is in general able to form chains that correctly represent the causal relationships between events in the text; rather, we are interested in the extent to which the predictions made by the inference component guide the system towards correct choices of specifications for anaphors.

The components of the system, namely the structure matcher, the AR rules, the inference mechanism and the generator, and also Boguraev's analyser, will be discussed in the approximate order in which they are applied.

(1) Boguraev's analyser usually performs quite reliably on the range of sentences with which it is equipped to deal; inappropriate choices of word sense or structure are fairly rare, except sometimes where pronouns are concerned. This is because pronouns differ from other noun phrases both semantically (in having very general meanings) and syntactically (for example, definite pronouns do not usually accept postmodification), and not all such differences were taken into account when the analyser was built. The working of the analyser's semantic routines is too complex to allow all the relevant problems to be removed without significant effort. As explained in 4.3.3, a few readings that are obviously incorrect therefore have to be removed or altered by hand before being input to SPAR; however, no theoretical problems of any relevance to anaphor resolution are sidestepped in so doing.

The inability of the analyser to deal properly with conjunctions means that the behaviour of anaphors in sentences involving conjunctions has not been investigated. In SPAR, all anaphors in a sentence are firmly resolved, and the focus registers updated, before the next sentence is considered; it would be interesting to see how this behaviour had to be modified in the presence of conjoined constituents of various kinds.

Finally, the lack of a completely reliable mapping between the case names used in dependency structures (AGENT, RECIPIENT etc.) and those in word-sense formulas (SUBJ, FOR etc.) makes the construction of text model current fragments rather more complex than it would otherwise be. This problem is worsened by the fact that the analyser assumes that the semantic preference restrictions on particular *syntactic* roles are to be found in particular *semantic* cases in formulas; thus the restriction on the direct object in an active sentence is taken as the filler of the OBJE role in the relevant formula even for a verb whose direct object plays some other semantic role.

However, apart from the problems listed here, the analyser generally worked reliably and presented few difficulties.

(2) SPAR's text representation and the corresponding structure matching apparatus allow anaphors and candidates to be compared moderately easily, with or without the information constraint in operation. However, the representation suffers from a lack of canonicality, which makes structure matching more difficult and time-consuming both for candidate assessment and for inference. This can largely be traced to the use, referred to above, of cases in formulas to represent syntactic as well as semantic information. But problems also arise when an entity is introduced in one way and then referred to in another, as in

(10-1) John let Bill borrow his car. Bill was grateful for the favour.

What appears to be needed here is not so much a canonical representation as a means to compare descriptions which describe an entity at different levels or from different perspectives. This requires a richer representation, perhaps involving a multiple-perspective device such as that of KRL rather than a single conceptual hierarchy. However, to construct such a representation might involve more inference than a shallow processing system could be expected to perform.

The separate and parallel treatment of each reading delivered by the analyser causes some inefficiency; pairs of readings often differ only slightly and/or in ways irrelevant to anaphor resolution, so that a great deal of processing is duplicated. However, an alternative approach, perhaps allowing uncertainty in the representation (so that a TMN node might be a Specialisation: of several alternative word senses, for example) would involve a large increase in the complexity of almost every part of the system. A still more radical (and more complex) alternative would be to integrate anaphor resolution with the operation of the analyser, resolving at least some anaphors fully or partially before parsing is complete. The extra semantic information provided by such resolution would enable some incorrect analysis paths to be pruned earlier than is currently the case. However, because, as I have argued, anaphor resolution is a largely collective process, and depends on the context provided by the whole sentence, the number of reliable early resolutions would probably not be very great; and in any case, the analyser's use of available semantic information at noun phrase and clause boundaries means that the parsing process is already reasonably efficient.

In the area of SPAR's word sense representation, the use of the THIS primitive and of senses as well as primitives in formulas makes it quite easy to define new senses economically and consistently when extending the lexicon. What consistency problems do arise are largely due to the use (forced by the analyser) of Wilks' syntax for formulas, rather than the "intermediate form" introduced in chapter 4. As was argued there, Wilks' syntax encourages considerable vagueness and implicitness which often conflicts with accuracy.

Thus the main drawback of the representational formalism is its non-canonicality. Otherwise it appears quite well suited to anaphor resolution.

(3) The Sidnerian anaphor resolution rules, modified and extended to cater for intrasentential candidates and to include the arbitrator and, where necessary, the post-inference collective heuristics, for the most part work reliably and efficiently. The order in which the AR rules suggest candidates is usually intuitively sensible. However, their use of different candidate orderings for

agent and non-agent animate pronouns is counterintuitive, and sometimes causes problems. It also made the extension of the rules to incorporate intrasentential candidates a little awkward.

Further, the distinction between candidate plausibility and implausibility is not always as cut and dried as the form of the AR rules presupposes. For example, in

(10-2) John walked over to the trees. He began to speak to them.

the preference of "speak" for an animate recipient would cause SPAR to reject the trees as referent of "them", and in the majority of cases it would be right to do so. Here, however, no other referent is available, so the plausibility requirement should be relaxed to allow the preference-breaking candidate to succeed.

(4) The inference mechanism is, when invoked, the most computationally expensive part of the system, and the one that least often produces useful predictions. The lack of predictions is largely due to the fairly small amount of world knowledge (only twenty or so inference rules) that are available. However, when predictions are made and are subsequently accepted by the arbitrator, they are almost always correct ones; thus the inference mechanism is a reasonably reliable part of the system. But its contribution to the system's performance is comparatively small: for example, inference chains were used to determine the resolution of only 12% of the pronouns in the stories in appendix B. (A large number of other chains were formed but either only confirmed resolution decisions already made or were rejected as incompatible with the predictions of other components of the system).

The reason that such a small number of rules have been provided is threefold. Firstly, the system is usually able, on the stories tested, to arrive at correct decisions even when the inference mechanism does not have the necessary world knowledge to construct chains. Secondly, there is a gratifying tendency for existing rules to give rise to correct chains for texts other than the one for which they were originally constructed. Thus, for example, several inference rules contain knowledge about making promises, and these rules formed useful chains in several stories involving promises. Thirdly, many useful inferences are object-oriented (see 8.4.1), only making use of linguistic information in the WSN and not of explicit inference rules.

All three of these tendencies provide some support for a shallow processing approach. They suggest that on many occasions, anaphors can be resolved without inference; that when inference is needed, a limited amount of knowledge about a concept can be useful in many different situations; and that linguistic knowledge as well as world knowledge can be very useful. However, the number and complexity of the texts processed are too small for these conclusions to be other than very tentative.

(5) The basic structure of the generator proves to be appropriate for producing paraphrases that satisfy the criteria given at the beginning of chapter 9. Sensible decisions are usually made about what information to include and how to order it. However, some problems are encountered at a more local level; for example, the selection of appropriate determiners and tenses is somewhat ad-hoc, making use of the "linguistic features" attached to TMN nodes. In chapter 4 it was noted that many of these features are themselves rather ad-hoc and would ideally be replaced in a cleaner version of

the representation.

10.3 Shallow processing for anaphor resolution

Given this assessment of SPAR's components, we can now ask more generally what evidence the system's performance provides for the feasibility of a shallow processing approach to resolving anaphors for various tasks and in various types of text.

The texts SPAR has processed fall into two classes. The first class, shown in appendix A, consists of forty short texts of one to three sentences which I have written to test or illustrate specific aspects of SPAR's processing; all the anaphors in them are resolved correctly. These texts give an indication of the range of phenomena that the system can handle. However, because they are specially constructed, they do not constitute direct evidence for the shallow processing hypothesis. They perhaps provide some indirect support for a shallow processing approach in the sense that they show that a shallow processing system can, with the limited amount of effort available in a project of this type, be made to cover quite a wide spectrum of phenomena.

The second class, shown in appendix B, consists of twenty-three texts written by people with little or no knowledge of SPAR's processing strategy; many were originally written for other language-processing systems. Some were written to fall within the analyser's grammatical coverage; some were not, but were edited to make them do so. These stories are on average nine sentences long; the longest has twenty-three sentences. Of the 242 pronouns in them, about[1] 226 (93%) are resolved correctly; of the 80 non-pronominal anaphors, 66 (82%) are resolved correctly. The figure for pronouns could almost certainly be be increased to 232 (96%) by the implementation of the error recovery procedures described below.

These figures are obviously of limited significance because of the simplicity of the texts processed compared to "real" texts, and because it was often necessary, in order to keep the complexity of incidental problems within reasonable bounds, to define new word senses with only the minimum of detail. However, it was encouraging to note that once the components of the system had been developed to the stage reported in this thesis, new stories did not often show up serious inadequacies in the program, and it was hardly ever necessary to extend existing word sense formulas to process new stories. In other words, the heavy use of word sense information by all parts of the system, including the inference component, did not lead to inappropriately large numbers of facts being stored as definitional (lexical) knowledge.

---

[1]These figures are only approximate because it is not always possible to decide unequivocally whether a particular phrase is anaphoric or not and, if it is, whether the specification selected for it is correct. However, the inaccuracies are unlikely to be more than one or two per cent.

The figures do not include the anaphors in text B23, which is significantly different from B22 only where generation, and not interpretation, is concerned.

Some attempt was made to test the system's robustness with variations on a single text by replacing one or more events with subtly but crucially different ones involving the same concepts: see e.g. texts A24-26, A37-40 and B1 and B9 in appendices A and B. Even so, there is no doubt that SPAR would make wrong decisions on some variations of many of the correctly-processed texts shown in appendix B. But tests of this type would not necessarily prove anything very interesting. The assumption behind the shallow processing hypothesis is that speakers or writers construct texts considerately; that is, they choose their words so that the applications of different types of knowledge by the reader tend to confirm rather than contradict one another. It is arguable that many of the variations that would cause SPAR (but not people) to misinterpret anaphors would do so precisely because they are less considerate than the original texts.

The pattern of the errors that SPAR did make when all the components were functioning as intended was interesting: few of the errors misled the system sufficiently to cause further misresolutions in subsequent sentences. Some possibilities for eliminating such "knock-on" effects as do occur are discussed below.

What evidence is there, then, that a shallow processing approach might be applicable to more than the particular text type (simple stories) and task (paraphrase) for which SPAR is designed?

Although SPAR was designed to process stories describing sequences of events, it also proved able to handle, with only trivial extensions, some texts of other types which had been written to test other text processing systems; see for example texts B21 and B22 in appendix B.[2] Indeed, the only part of the system that is specific to stories is the inference component. For non-story texts, the inference component might be generalised to look for coherence relations rather than causal chains, perhaps making use of the insights of Hobbs [1979] (see 5.2.2).

Shallow processing using the techniques described in this thesis might be a promising basis for anaphor resolution in interactive tasks such as natural language database query. When SPAR does make wrong decisions, they are usually the result of predictions made by the less reliable post-inference components of the system. In such cases, instead of applying these components, the system could ask the user to indicate the correct decision; hopefully this would not occur so often as to degrade the system's usefulness. Furthermore, anaphor resolution by shallow processing methods is an attractive possibility for machine translation applications where only a rough translation is required or where post-editing is performed. Here again, doubtful resolutions could be indicated if desired.

Perhaps the most serious limitation of the significance of the work presented here is that it is inevitably concerned with texts that are far simpler than those occurring in "real" applications. However, this criticism can be levelled

---

[2]For B21, the system only needed to be extended to handle FDNP anaphors that specified sets of entities not previously specified by a single phrase. The analogous capability for pronouns was already present. For B22, no extensions were needed.

at much research in natural language processing, and there is reason to be comparatively optimistic about the possibilities of "scaling up" the techniques presented here, because of their reliance on linguistic rather than world knowledge. In the next few years, reasonably comprehensive computational grammars of English should become available; and on the semantic front, the existence of on-line dictionaries means that the required information about word meanings is potentially available. Research such as that of Alshawi [1985] offers the hope that this semantic information can be converted into a computationally exploitable form that could play an analogous role in anaphor resolution to the role played in SPAR by Wilksian word-sense formulas.

10.4 Directions for further work

This final section contains suggestions for extending the SPAR system and the ideas it is based on. Perhaps the most serious gaps in SPAR as it stands are the lack of any treatment of quantifier scope and of any mechanism for recovering from the errors that a shallow processing system will inevitably make. Some possible ways of filling these gaps are therefore discussed here. We will then look at the possibility of extending SPAR's coverage to other types of anaphoric and non-anaphoric ambiguities, and will conclude the thesis with some speculations on the longer-term future of a shallow processing approach to ambiguity resolution.

10.4.1 Intensional contexts and quantifier scope

In chapter 4 it was claimed that the "context of identity" information attached to TMN nodes could be extended to make SPAR sensitive to quantifier scope as well as intensional contexts. This might be done as follows.

The sentence

(10-3) Bill wants to break <u>a window</u>.

is ambiguous between readings which might be represented semi-formally[3] as

(L10-3a) ∃x:window [wants(Bill,break(Bill,x))]

("there is a specific window, and Bill wants to break it") and

(L10-3b) wants(Bill, ∃x:window [break(Bill,x)])

("Bill wants to break some window - any window will do"). Reading (L10-3b) is more plausible in a neutral context.

---

[3]Because of its familiarity, predicate logic will be used to express semantic distinctions in this section. However, the notion of "context of identity" is perhaps more closely related to Kamp's concept of a discourse representation than to anything in predicate logic. Furthermore, we saw in chapter 2 that Kamp's formalism is able to deal naturally with some phenomena that are problematical to a predicate logical approach. Any development of SPAR to deal with quantifier scope would therefore probably use Kamp's framework rather than standard logic. The two frameworks are effectively isomorphic as far as the examples used here are concerned.

In SPAR's representation of (10-3), the TMN node for "window" would have a context of identity such as (WANT1-1), showing that the window probably only has a specific identity in the context of Bill's desires (thus favouring (L10-3b)).

Similar ambiguities can be caused by certain quantifiers. For example

(10-4) Every farmer owns a donkey.

may in principle be interpreted either as

(L10-4a) $\exists$d:donkey [$\forall$f:farmer [owns(f,d)]]

or as

(L10-4b) $\forall$f:farmer [$\exists$d:donkey [owns(f,d)]]

(although the semantics of ownership in fact makes (L10-4b) more likely in the null context). Thus, by analogy with (10-3), it would seem sensible to give the TMN node for "a donkey" the context of identity (FARMER1-1): the donkey only has a specific identity once the choice of farmer is made.

Resolution of anaphors in subsequent sentences would both be constrained by contexts of identity and would help to determine them more fully. Thus in

(10-5) [1] Every farmer owns a donkey.
[2] He beats it.

the pronouns would force [2] to be interpreted within the (FARMER1-1) context of identity and confirm the dependence of DONKEY1-1 on FARMER1-1 for its identity. On the other hand the resolution of the underlined pronoun in the (somewhat forced) text

(10-6) [1] Every farmer owns a donkey.
[2] Every factory worker owns it too.
[3] That's one of the advantages of collectivisation.

would cause [2] to be interpreted in the null context of identity, and rule out the dependence of DONKEY1-1 on FARMER1-1. Similarly, one can imagine continuations to (10-3) which would either confirm or rule out the dependence of WINDOW1-1 on WANT1-1.

Such a treatment of intensional contexts and quantifier scope would have the advantage of allowing logically distinct readings to be represented by the same structure and only resolved later when necessary (and possible). Hobbs [1983] argues that since many quite comprehensible sentences have a large number of different scopings, a "scope-neutral" representation is desirable; and VanLehn [1978] maintains that people normally do not disambiguate quantifier scope when they understand sentences. In addition, VanLehn shows that the determination of quantifier scope (when it is carried out) is a complex process depending on syntactic, semantic and pragmatic factors. There therefore seems no reason to attempt it in a shallow processing system before enough information is available to force a decision.

The ideas presented in this brief outline are necessarily simplified, and it should not be imagined that the interactions between contexts of identity and anaphor resolution would in practice be at all straightforward. Indeed the process would in general appear to be a collective one involving the satisfaction of as many preferential constraints as possible. Specifying the nature and strength of these constraints would be a complex task.

10.4.2 Recovering from errors

A shallow processing system such as SPAR can be expected to make a certain proportion of errors in resolving anaphors, especially where decisions are uncertain, i.e. determined by the results of inference, by subsequently-applied collective heuristics, or (especially) by the weak preferences of the AR rules. In addition, and independent of shallow processing considerations, Mellish [1982] argues that there is no arbitrary point (end of noun phrase, end of sentence, etc.) at which enough information will necessarily have accumulated to resolve anaphors reliably. At present, however, SPAR always resolves anaphors sentence by sentence, and has no means of detecting or correcting any incorrect resolutions it makes.

Thus some of SPAR's errors are due to its being a shallow processing system, and some are due to its resolving all ambiguities irrevocably at the end of each sentence. We will now see how some errors of the second type might be eliminated.

Two approaches to the problem are conceivable: a breadth-first approach and a depth-first, backtracking approach. In a breadth-first approach, uncertain decisions would be postponed, and alternative versions of context carried along in the hope that later processing would reveal one version, and therefore one choice of referent (or indeed sense or structure), as preferable. For example, just as in 6.1 we saw how SPAR sometimes prefers one reading of a sentence to another on the grounds that it requires fewer AR rule applications, so a version of *context* that allowed anaphors to be resolved more easily would also be preferred.

However, since many of SPAR's decisions are uncertain, a breadth-first approach would explode quite rapidly as more sentences were processed. It would therefore probably only be feasible to postpone uncertain decisions in this way for a sentence or two at most. Thereafter, a backtracking approach, in which the system recognises that an error has occurred and goes back to repair it, would be necessary, either on its own or in addition to limited breadth-first processing. This raises the question of how the system could (a) recognise that an erroneous decision has been made and (b) identify which decision of the many that have been made is the culprit.

Misresolutions of anaphors can be expected sometimes to give rise to apparent violations of the requirements of referential adequacy and efficiency (see 9.1.2) which the writer of the text is assumed to observe. These apparent violations could be used to identify and correct earlier errors.

The requirement of referential adequacy will appear to be broken if the only referent(s) that the AR rules can suggest for a pronoun are ruled out by configurational constraints. When this occurs, the pronoun will appear to the program to be referentially inadequate because it fails to identify a referent. For example, if SPAR decided (incorrectly) when processing

(10-7) [1] Mary wanted to give Susan a present.
[2] She went to a shop which sold computers.
[3] She walked over to one.
[4] She decided that it would be a good present for Susan.

that the "she" in [2] was Susan and not Mary, then Susan would become discourse focus and be selected as referent of "she" in [3] as well. However, in

[4], problems would arise because "she" c-commands "Susan" and therefore cannot corefer with it. Mary is by now not in focus at all and will never be suggested by the AR rules.

The correct action here would be to search backwards through a record of the anaphor resolution decisions made earlier, and provisionally alter the most recent *uncertain* decision in which the referent involved in the c-command clash (here, Susan) was preferred. In this case, the most recent such decision is the choice of Susan rather than Mary as referent of "she" in [2]. If this decision is reversed and the story is re-processed from that point on, Mary will become discourse focus and no problems will arise.

In cases where backtracking of this type fails to resolve the problem, the AR rules could be forced to look outside the focus registers for specifications for pronouns, using simple recency criteria to decide between candidates.

If these ideas were implemented in SPAR, the system would correctly process both of the texts in appendix B (B6 and B10) where it currently ignores a sentence because of an "unresolvable" pronoun. This would raise the proportion of correctly resolved pronouns in appendix B to about 96%.

An apparent violation of the requirement of referential efficiency - for example, if a FDNP is used where a pronoun would be unambiguous - can also signal an earlier wrong decision. Grosz, Joshi and Weinstein's centering rule (see 5.4.4) could perhaps be used to detect such violations. The appropriate action here is similar to the referential inadequacy case: backtrack to the most recent uncertain decision in favour of the referent in question, and reverse it. Suppose, for example, that in

(10-8) [1] Mary wanted to give Susan a present.
[2] She went to a shop which sold computers.
[3] She bought one.
[4] Susan loved it.

the "she" in [2] is again misresolved as Susan, so that Susan is also chosen as referent of "she" (and therefore as discourse focus) in [3]. The use of "Susan" in [4] will then appear to be referentially inefficient; "she" would have identified Susan just as well. If the system backtracks to the most recent uncertain decision in favour of Susan, i.e. that in [2], and reverses it, then Mary will be discourse focus when [4] is encountered and all will be well.

Errors in inference could sometimes be detected and corrected by applying Granger's [1980] principle that "the best context inference is the one which accounts for the most actions of a story character". Granger's system ARTHUR applied this principle to understand texts where an initially plausible explanation later turns out to be wrong. In

(10-9) [1] Mary picked up a magazine.
[2] She swatted a fly.

the initial inference that Mary planned to read the magazine was replaced, when [2] was read, by the inference that she picked up the magazine to use it as a weapon.

In SPAR, the equivalent of Granger's principle would be that single-tendril inference chains that explain a single assertion should be replaced by chains connecting one assertion to another. Thus in

(10-10) [1] John wanted Bill to mend his car.
[2] Bill had promised to lend it to him.

an initial single-tendril chain, to the effect that the car is John's because John is likely to want his own possessions to work, would be overturned, when [2] is processed, by the more complex but more explanatory chain saying that John wanted Bill's car mended because he was going to borrow it.

10.4.3 Extending SPAR's linguistic coverage

At present, the range of linguistic ambiguities that SPAR is equipped to deal with thoroughly is fairly limited. Lexical and structural ambiguities are only properly tackled when they interact with anaphora as described in 6.1; and a cursory glance at Halliday and Hasan [1976] will reveal many kinds of anaphora that SPAR does not handle, perhaps the most notable being ellipsis and "do so" anaphors.

The "marker passing" approach to lexical ambiguity resolution is consistent with the shallow processing emphasis on linguistic knowledge. Marker passing was originally proposed by Quillian [1968], and has been used for word-sense disambiguation by Alshawi [1983] and Hirst [1983] among others. In such approaches, the use of a word sense effectively increases the activation level of semantically related senses, and readings containing more activated senses are preferred. Wilksian word sense formulas, when allowed to contain other word senses, would appear to provide a sufficiently rich network for many if not most lexical ambiguities to be resolved by marker passing.

Any practically useful system that resolved anaphors by shallow processing means would clearly have to have wider coverage of types of anaphora than SPAR does. To give SPAR a wider coverage, developments in various parts of the system, most notably the meaning representation and the AR rules, would be required. Such extensions would provide further evidence of the usefulness of some of the techniques used and developed in this thesis, but would not on their own necessarily cast any more light on the feasibility of a shallow processing approach to anaphora. For example, there is no obvious reason why shallow processing should be suitable for definite pronouns but not for "do so" anaphors. Thus to extend SPAR to process "do so" anaphors, but only in texts of the same complexity as those currently handled, would be far less impressive than to extend it to deal with more realistically complex texts, even if those texts were restricted to containing only nominal anaphors.

Rather, what is needed in the long term is extensions of the system along many different dimensions together to enable it to deal with more complex texts. Work along these lines would encounter many of the unsolved problems of natural language processing; however, it is to be hoped that other unsolved problems, notably those of complex inference, would not be so severe in a shallow processing context. If so, then at least for some language-oriented tasks such as translation, the goal of genuinely useful automatic processing of non-specialised natural language texts will be that much easier to achieve.

Appendix A: Short example texts and paraphrases

The texts in this appendix test and illustrate particular aspects of SPAR's processing. Except where otherwise stated, they were written by the present author. To allow easier comparison, SPAR's output for each input sentence is shown immediately below the input. All material in THIS FONT is system input and output. Bracketed section numbers show where the phenomenon illustrated by the example is discussed in the main body of the thesis.

1. The "principle of anaphoric success" (6.1.1, 6.1.3).

Structural ambiguity can be resolved by considering what entities are available in context. Compare the interpretation of [3] in A1 and A2 below. (The often inaccurate use of NEAR to express all kinds of spatial relationship is due to the analyser's treatment of all locative prepositions with the same LOCATION case).

A1 [1]. JOHN PUT A BOWL ON THE FLOOR.
NEAR A FLOOR, JOHN PUT A BOWL.

[2]. HE PICKED UP SOME BISCUITS.
JOHN PICKED UP SOME BISCUITS.

[3]. HE PUT THE BISCUITS IN THE BOWL ON THE FLOOR.
NEAR THE BOWL, JOHN PUT THE BISCUITS.

A2 [1]. JOHN PICKED UP SOME BISCUITS.
JOHN PICKED UP SOME BISCUITS.

[2]. HE PUT THEM IN A BOWL.
NEAR A BOWL, JOHN PUT THE BISCUITS.

[3]. HE PUT THE BISCUITS IN THE BOWL ON THE FLOOR.
NEAR A FLOOR, JOHN PUT THE BISCUITS.

Word sense ambiguity can also be resolved by accepting the more focussed option.

A3 [1]. JOHN PUT THE CARDS ON THE TABLE.
NEAR A DINING TABLE, JOHN PUT SOME CARDS.

[2]. HE PUT THE TOOLS ON THE FLOOR.
NEAR A FLOOR, JOHN PUT SOME REPAIR IMPLEMENTS.

[3]. HE PICKED UP THE JACK.
JOHN PICKED UP THE LIFTING DEVICE WHICH WAS AMONG THE REPAIR IMPLEMENTS WHICH JOHN PUT NEAR THE FLOOR.

A4 [1]. JOHN PUT THE DOG ON THE TABLE.
NEAR A DINING TABLE, JOHN PUT A DOG.

[2]. HE EXAMINED ITS LEGS.
JOHN EXAMINED THE DOG'S LIMBS.

2. FDNP resolution (6.2.1).

A5 illustrates cospecification [2] and associated specification [3].

A5 [1]. JOHN HAS BOUGHT A NEW JAGUAR.
JOHN HAS PURCHASED A NEW JAGUAR.

[2]. THE CAR IS ULTRA-MODERN IN EVERY RESPECT.
THE JAGUAR IS ULTRA-MODERN IN EVERY RESPECT.

[3]. THE INDICATORS ARE CONTROLLED BY A MICROPROCESSOR.
THE JAGUAR'S INDICATORS ARE CONTROLLED BY A MICROPROCESSOR.

The establishment of cospecification may use information from the sentential context of the antecedent as well as the antecedent itself. In A6, but not in A7, "his assailant" is identified with the "man".

A6 [1]. A MAN ATTACKED JOHN IN THE PARK.
AT A PARK, A MAN ATTACKED JOHN.

[2]. HE RAN AWAY FROM HIS ASSAILANT.
JOHN FLED FROM THE MAN.

A7 [1]. A MAN APPROACHED JOHN IN THE PARK.
AT A PARK, A MAN APPROACHED JOHN.

[2]. HE RAN AWAY FROM HIS ASSAILANT.
THE MAN FLED FROM THAT MAN'S ASSAILANT.

The final sentences in A8 and A9 illustrate inferred specification (i.e. relaxing the information constraint).

A8 [1]. JOHN WAS DRIVING ALONG THE MOTORWAY.
JOHN WAS DRIVING NEAR A MOTORWAY.

[2]. HIS JAGUAR BROKE DOWN.
JOHN'S JAGUAR WHICH JOHN WAS DRIVING NEAR THE MOTORWAY IN STOPPED WORKING.

A9 [1]. THE HEIRESS LIVED AS A RECLUSE.
AN HEIRESS RESIDED LIKE A RECLUSE.

[2]. SHE DIED UNDER MYSTERIOUS CIRCUMSTANCES.
THE HEIRESS DIED UNDER MYSTERIOUS CIRCUMSTANCES.

[3]. THE MURDERER WAS NOT FOUND.
THE PERSON WHO KILLED THE HEIRESS WAS NOT FOUND.

The information constraint may be broken, in direct cospecification, by modifiers (A10) but not by head nouns (A11).

A10 [1]. THE HEIRESS LIVED AS A RECLUSE.
AN HEIRESS RESIDED LIKE A RECLUSE.

[2]. NOBODY SAW THE OLD WOMAN.
NOBODY SAW THE HEIRESS.

A11 [1]. THE OLD WOMAN LIVED AS A RECLUSE.
AN OLD WOMAN RESIDED LIKE A RECLUSE.

[2]. NOBODY SAW THE HEIRESS.
NOBODY SAW AN HEIRESS.

Indefinite as well as definite noun phrases may by resolved as anaphors.

A12 [1]. JOHN ENTERED A RESTAURANT.
JOHN ENTERED A RESTAURANT.

[2]. A WAITER CAME TOWARDS HIM.
A WAITER WHO WORKED AT THE RESTAURANT CAME TO JOHN.

3. Resolving descriptional anaphors (6.2.2).

When resolving phrases with heads "one" and "some", set membership is preferred where possible (A13, A14, A15); otherwise sense alone is shared (A16).

A13 [1]. JOHN PICKED SOME FLOWERS.
JOHN PLUCKED SOME FLOWERS.

[2]. MARY ASKED HIM FOR THE RED ONE.
FROM JOHN, MARY DEMANDED THE RED FLOWER WHICH WAS AMONG THE FLOWERS WHICH JOHN PLUCKED.

A14 [1]. JOHN PICKED SOME FLOWERS.
JOHN PLUCKED SOME FLOWERS.

[2]. MARY ASKED HIM FOR A RED ONE.
FROM JOHN, MARY DEMANDED A RED FLOWER WHICH WAS AMONG THE FLOWERS WHICH JOHN PLUCKED.

A15 [1]. JOHN PICKED SOME FLOWERS.
JOHN PLUCKED SOME FLOWERS.

[2]. MARY ASKED HIM FOR ONE.
FROM JOHN, MARY DEMANDED ONE OF THE FLOWERS WHICH JOHN PLUCKED.

A16 [1]. JOHN LIKED BANANAS.
JOHN LIKED BANANAS.

[2]. HE ASKED MARY FOR ONE.
FROM MARY, JOHN DEMANDED A BANANA.

The "one" in [2] below is parasitic; its resolution depends on that of "them".

A17 [1]. WENDY EXAMINED THE T-SHIRTS ON THE TABLE.
WENDY EXAMINED THE T-SHIRTS WHICH WERE NEAR A DINING TABLE.

[2]. SHE PICKED UP ONE OF THEM.
WENDY PICKED UP ONE OF THE T-SHIRTS WHICH WERE NEAR THE DINING TABLE.

4. Plural pronoun resolution (6.2.2).

The correct resolution of "them" in [2] below depends on the selection of a set of semantically similar entities, i.e. John and Mary rather than John, Mary and the park.

A18 [1]. JOHN MET MARY IN THE PARK.
AT A PARK, JOHN MET MARY.

[2]. A POLICEMAN SAW THEM.
A POLICEMAN SAW JOHN AND MARY.

5. "Normal mode" inference (6.2.3).

Extraction matching is inappropriate for normal mode inference (A19); however, semantic constraints should be applied (A20-21). A20 is adapted from Wilks [1975a].

A19 [1]. BILL STAYED AT HOME.
BILL STAYED AT HOME.

[2]. JOHN WENT TO LONDON.
JOHN WENT TO LONDON.

[3]. HE TRAVELLED FROM BIRMINGHAM TO CAMBRIDGE.
FROM BIRMINGHAM, JOHN TRAVELLED TO CAMBRIDGE.

A20 [1]. I BOUGHT THE WINE.
I PURCHASED SOME WINE.

[2]. I SAT ON A ROCK.
I SAT DOWN NEAR A ROCK.

[3]. I DRANK IT.
I IMBIBED THE WINE.

A21 [1]. THE MONKEYS PICKED SOME BANANAS.
SOME MONKEYS PLUCKED SOME BANANAS.

[2]. THEY WERE RIPE.
THE BANANAS WERE RIPE.

[3]. THEY ATE THEM.
THE MONKEYS DEVOURED THE BANANAS.

6. Augmenting the focus registers with intrasentential candidates (6.3.3).

An intrasentential (strict) anaphor is preferred to a potential focus (A22); however, an intrasentential cataphor is not (A23).

A22 [1]. MARY ARRIVED AT THE CLUB WITH JOHN.
WITH JOHN, MARY ARRIVED AT A CLUB.

[2]. SUSAN WAS TELLING BILL'S SISTER ABOUT HIS BEHAVIOUR.
TO BILL'S SISTER, SUSAN WAS COMMUNICATING BILL'S BEHAVIOUR.

A23 [1]. MARY ARRIVED AT THE CLUB WITH JOHN.
WITH JOHN, MARY ARRIVED AT A CLUB.

[2]. SUSAN WAS TELLING HIS SISTER ABOUT BILL'S BEHAVIOUR.
TO JOHN'S SISTER, SUSAN WAS COMMUNICATING BILL'S BEHAVIOUR.

7. Applying syntactic constraints (6.4.1).

The timely application of configurational constraints can remove the need for inference. A24 and A25 are adapted from Sidner [1979]. (A special hack was needed to recognize that the idiom "X bit Y in the hand/paw" means "X bit Y's hand/paw").

A24 [1]. I TOOK MY DOG TO THE VET ON FRIDAY.
TO A VET, AND ON FRIDAY, I CONVEYED MY DOG.

[2]. HE BIT HIM IN THE HAND.
AT THE VET'S HAND, THE DOG BIT THAT VET.

A25 [1]. I TOOK MY DOG TO THE VET ON FRIDAY.
TO A VET, AND ON FRIDAY, I CONVEYED MY DOG.

[2]. HE INJECTED HIM WITH A NEW MEDICINE.
WITH A NEW MEDICINE, THE VET INJECTED THE DOG.

A26 [1]. I TOOK MY DOG TO THE VET ON FRIDAY.
TO A VET, AND ON FRIDAY, I CONVEYED MY DOG.

[2]. HE BIT HIM IN THE PAW.
AT THE DOG'S PAW, THE VET BIT THAT DOG.

8. The c-commanded pronoun heuristic (6.4.3).

(See text B1 in the next appendix for the repetition heuristic, and B2 for focus retention). "John" c-commands "his" below, but "Bill" does not.

A27 [1]. ON HIS ARRIVAL JOHN REALISED THAT BILL WAS DRUNK.
JOHN REALISED ON JOHN'S ARRIVAL THAT BILL WAS DRUNK.

9. CSI rule application (8.2.1).

In A28 and A29 (adapted from Wilks [1975a] and [1975b] respectively), "them" is resolved by CSI chains. (See appendix C for more detailed examples of inference).

A28 [1]. THE SOLDIERS FIRED AT THE WOMEN.
SOME SOLDIERS FIRED AT SOME WOMEN.

[2]. WE SAW SEVERAL OF THEM FALL.
WE SAW SEVERAL OF THE WOMEN WHO THE SOLDIERS FIRED AT FALL.

A29 [1]. THE DOGS CHASED THE CATS.
SOME DOGS CHASED SOME CATS.

[2]. ONE OF THEM SQUEALED IN PAIN.
ONE OF THE CATS WHICH THE DOGS CHASED SQUEALED BECAUSE OF PAIN.

10. Negative matches during inference

...may lead to negative (A30) or positive (A31) inference chains (8.2.4).

A30 [1]. JOHN KILLED BILL.
JOHN KILLED BILL.

[2]. HE RAN AWAY.
JOHN FLED.

A31 [1]. THE SOLDIERS FIRED AT THE WOMEN.
SOME SOLDIERS FIRED AT SOME WOMEN.

[2]. THEY DID NOT FALL.
THE WOMEN DID NOT FALL.

11. Non-explanatory inference chains

...can still help to resolve pronouns (8.3.1).

A32 [1]. JOHN GAVE BILL A BANANA.
TO BILL, JOHN PRESENTED A BANANA.

[2]. HE ATE IT.
BILL DEVOURED THE BANANA.

A33 [1]. FRED GAVE JOHN A HAMMER.
TO JOHN, FRED PRESENTED A HAMMER.

[2]. HE HIT BILL WITH IT.
WITH THE HAMMER, JOHN HIT BILL.

12. The generator.

The generator tries to select minimal modifiers to distinguish the intended referent from others (9.3.1). In A34, "new" does the job in [3]; in A35, it does not, and a full relative clause is needed.

A34 [1]. JOHN SOLD HIS OLD CAR.
JOHN SOLD JOHN'S OLD CAR.

[2]. HE BOUGHT A NEW ONE.
JOHN PURCHASED A NEW CAR.

[3]. IT BROKE DOWN.

THE NEW CAR STOPPED WORKING.

A35 [1]. BILL BOUGHT A NEW CAR.
BILL PURCHASED A NEW CAR.

[2]. JOHN BOUGHT A NEW CAR.
JOHN PURCHASED A NEW CAR.

[3]. IT BROKE DOWN.
THE CAR WHICH JOHN PURCHASED STOPPED WORKING.

Sentence [2] below exercises the generator to the full (9.4.2).

A36 [1]. JOHN WENT TO THE ZOO.
JOHN WENT TO A ZOO.

[2]. HE SAW THE ELEPHANTS WAVING THEIR TRUNKS.
WHILE THE ELEPHANTS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO WAVED THOSE ELEPHANTS' PROBOSCES, JOHN SAW THOSE ELEPHANTS.

13. Some variations on the theme of "promising".

The pronouns here are resolved by a combination of CSI and semantic (word-sense) constraints.

A37 [1]. JOHN PROMISED BILL THAT HE WOULD MEND HIS CAR.
JOHN PROMISED BILL THAT JOHN WOULD REPAIR BILL'S CAR.

A38 [1]. JOHN PROMISED HIS FATHER THAT HE WOULD DO HIS HOMEWORK.
JOHN PROMISED JOHN'S FATHER THAT JOHN WOULD DO JOHN'S HOMEWORK.

A39 [1]. JOHN PROMISED HIS SON THAT HE WOULD DO HIS HOMEWORK.
JOHN PROMISED JOHN'S SON THAT JOHN WOULD DO THAT SON'S HOMEWORK.

A40 [1]. JOHN PROMISED HIS SON THAT HE WOULD RECEIVE A COMPUTER FOR CHRISTMAS.
JOHN PROMISED JOHN'S SON THAT BECAUSE OF CHRISTMAS, THAT SON WOULD RECEIVE A COMPUTER.

Appendix B: Longer example texts and paraphrases

The texts in this appendix were written by people with little or no knowledge of SPAR's processing strategies. Some were written to conform to the analyser's grammatical coverage; the others were edited to make them do so.

Aspects of the following story are discussed in 6.4.2, 6.4.3, 8.3.2 and 8.4.2. See appendix C for the inference involved in processing [1] to [3]. The repetition heuristic (see 6.4.3) led to the resolution of "he" in [4], "he" in [5] and "his" in [7].

B1 [1]. JOHN PROMISED BILL THAT HE WOULD MEND HIS CAR.
JOHN PROMISED BILL THAT JOHN WOULD REPAIR BILL'S CAR.

[2]. HE TOOK IT TO HIS FRIEND'S GARAGE.
TO JOHN'S FRIEND'S GARAGE, JOHN CONVEYED THE CAR.

[3]. HE TRIED TO PERSUADE HIS FRIEND THAT HE SHOULD LEND HIM SOME TOOLS.
JOHN ATTEMPTED TO CONVINCE JOHN'S FRIEND THAT THAT FRIEND SHOULD LOAN JOHN SOME REPAIR IMPLEMENTS.

[4]. HIS FRIEND SAID THAT HE WAS NOT ALLOWED TO LEND TOOLS.
JOHN'S FRIEND SAID THAT THAT FRIEND WAS NOT ALLOWED TO LOAN ANY REPAIR IMPLEMENTS.

[5]. JOHN ASKED HIS FRIEND TO SUGGEST SOMEONE FROM WHOM HE COULD BORROW TOOLS.
JOHN REQUESTED JOHN'S FRIEND TO RECOMMEND SOMEONE WHO JOHN COULD BORROW SOME REPAIR IMPLEMENTS FROM.

[6]. HIS FRIEND DID NOT ANSWER.
JOHN'S FRIEND DID NOT ANSWER.

[7]. FULFILLING HIS PROMISES WAS IMPORTANT TO JOHN.
DISCHARGING JOHN'S PROMISES WAS URGENT TO JOHN.

[8]. HE WAS ANGRY.
JOHN WAS ANGRY.

[9]. HE LEFT.
JOHN DEPARTED.

The focus retention heuristic is used for "he" in [5] below.

B2 [1]. JOHN WENT TO A RESTAURANT.
JOHN WENT TO A RESTAURANT.

[2]. HE ASKED THE WAITER FOR A CURRY.
FROM THE WAITER WHO WORKED AT THE RESTAURANT, JOHN DEMANDED A CURRY.

[3]. HE ATE IT.
JOHN DEVOURED THE CURRY.

[4]. HE PAID THE CASHIER.

JOHN PAID A CASHIER.

[5]. HE LEFT.
JOHN DEPARTED.

The following is adapted from Cater [1981]. See 8.2.1 and 8.2.3.

B3 [1]. BILL WENT TO THE ZOO WITH JILL.
WITH JILL, BILL WENT TO A ZOO.

[2]. THEY GAVE THE MONKEYS SOME BANANAS WHICH THEY ATE.
TO THE MONKEYS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO, BILL AND JILL PRESENTED SOME BANANAS, AND THE MONKEYS DEVOURED THE BANANAS.

[3]. THEY WENT TO THE RESTAURANT.
BILL AND JILL WENT TO A RESTAURANT.

[4]. THEY DRANK SOME TEA.
BILL AND JILL IMBIBED SOME TEA.

[5]. JILL TOOK BILL'S MONEY.
JILL TOOK HOLD OF BILL'S MONEY.

[6]. SHE GAVE IT TO THE TRAMP WHO WAS TALKING TO THEM.
TO THE TRAMP WHO WAS TALKING WITH BILL AND JILL, JILL PRESENTED THE MONEY.

In the following story, note that:

(1) The repetitiveness in [2] is due to the generator not recognizing that "John and Mary lived in the garden" means the same as "Mary lived with John in the garden".

(2) The pseudo-preposition "outof" appears in the input because the analyser's grammar cannot deal with multiple prepositions.

B4 [1]. MARY LIVED WITH JOHN.
MARY RESIDED WITH JOHN.

[2]. THEY LIVED IN THE GARDEN WHICH GOD HAD GIVEN THEM.
JOHN AND MARY RESIDED AT THE GARDEN WHICH MARY RESIDED WITH JOHN AT AND WHICH GOD HAD PRESENTED TO JOHN AND MARY.

[3]. HE APPEARED IN IT.
GOD APPEARED AT THE GARDEN.

[4]. HE GAVE THEM A GOLD BOX.
TO JOHN AND MARY, GOD PRESENTED A GOLD BOX.

[5]. HE TOLD THEM THAT THEY COULD NOT TAKE THINGS OUTOF IT.
GOD INFORMED JOHN AND MARY THAT FROM THE BOX, JOHN AND MARY COULD NOT TAKE HOLD OF ANY THINGS.

[6]. THEY OBEYED HIS WORD FOR MANY YEARS.
FOR MANY YEARS, JOHN AND MARY OBEYED GOD'S SPOKEN WORD.

[7]. A HORSE APPEARED.
A HORSE APPEARED.

[8]. IT WAS BAD.
THE HORSE WAS BAD.

[9]. IT TOLD THEM TO BREAK INTO THE BOX.
THE HORSE ORDERED JOHN AND MARY TO BURGLE THE BOX.

[10]. THEY OBEYED IT.
JOHN AND MARY DID AS THE HORSE WANTED.

[11]. MONEY CAME OUTOF THE BOX.
SOME MONEY CAME FROM THE BOX.

[12]. NARCOTICS CAME OUTOF THE BOX.
SOME NARCOTICS CAME FROM THE BOX.

[13]. OTHER BAD THINGS CAME OUTOF THE BOX.
SOME OTHER BAD THINGS CAME FROM THE BOX.

[14]. GOD SAW THIS.
GOD SAW THE THINGS COME FROM THE BOX.

[15]. HE THREW THEM OUTOF HIS GARDEN.
FROM GOD'S GARDEN, GOD THREW JOHN AND MARY.

The following text presents few problems.

B5 [1]. MARY DECIDED TO GO TO PARIS.
MARY RESOLVED TO GO TO PARIS.

[2]. IT IS A BIG CITY.
PARIS IS A BIG CITY.

[3]. SHE BELIEVED THAT THE SHOPS WOULD BE INTERESTING.
MARY BELIEVED THAT SOME SHOPS WOULD BE INTERESTING.

[4]. THEY WOULD HAVE CHEAP THINGS FOR SALE.
THE SHOPS WOULD INCLUDE THE CHEAP THINGS WHICH WERE FOR SALE.

[5]. SHE WOULD GET CLEVER BOOKS.
MARY WOULD OBTAIN SOME WELL-ARGUED BOOKS.

[6]. SHE WOULD BUY RED FLOWERS.
MARY WOULD PURCHASE SOME RED FLOWERS.

[7]. THEY WOULD BE FUN.
THE FLOWERS WOULD BE FUN.

The next story also presents few problems until the final sentence, where "her" is unresolvable because Mary is no longer focussed enough to be suggested by the AR rules (but see 10.4.2).

B6 [1]. JOHN WENT TO THE GARDEN.
JOHN WENT TO A GARDEN.

[2]. THERE WERE FLOWERS IN IT.
SOME FLOWERS EXISTED AT THE GARDEN.

[3]. MARY SAW HIM.
MARY SAW JOHN.

[4]. SHE RAN AFTER HIM.
MARY SPRINTED TO JOHN.

[5]. HE WAS EATING BANANAS.
JOHN WAS DEVOURING SOME BANANAS.

[6]. SHE ASKED HIM FOR ONE.
FROM JOHN, MARY DEMANDED ONE OF THE BANANAS WHICH JOHN WAS DEVOURING.

[7]. HE GAVE HER ONE.
TO MARY, JOHN PRESENTED ONE OF THE BANANAS WHICH JOHN WAS DEVOURING.

[8]. THEY WENT TO THE FOUNTAIN.
JOHN AND MARY WENT TO A FOUNTAIN.

[9]. JOHN JUMPED INTO IT.
JOHN JUMPED TO THE FOUNTAIN.

[10]. HE FELT GLAD.
JOHN EXPERIENCED A GLAD FEELING.

[11]. MARY WAS GLAD TO SEE HIM PLAYING.
MARY WAS GLAD BECAUSE WHILE JOHN GAMBOLLED, MARY SAW JOHN.

[12]. SHE PICKED SOME FLOWERS.
MARY PLUCKED SOME FLOWERS.

[13]. SHE THREW THEM AT HIM.
TO JOHN, MARY THREW THE FLOWERS WHICH MARY PLUCKED.

[14]. HE CAUGHT THEM.
JOHN CAUGHT THE FLOWERS WHICH MARY PLUCKED.

[15]. HE TOLD HER THAT HE HAD LEFT SCHOOL.
IGNORED BECAUSE OF UNRESOLVABLE PRONOUN.

The following text is about as long as can be handled accurately without a global focussing mechanism. The slightly odd paraphrases of the last two sentences are due to SPAR not recognizing that they are generic.

B7 [1]. THE BIG LORRY CAME TO THE ZOO.
A BIG LORRY CAME TO A ZOO.

[2]. VARIOUS ANIMALS WERE IN THEIR CAGES.
VARIOUS ANIMALS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO WERE AT THOSE ANIMALS' CAGES.

[3]. SOME CONTAINED SHEEP.
SOME OF THE ANIMALS' CAGES CONTAINED SOME SHEEP WHICH WERE AMONG THOSE ANIMALS.

[4]. SOME CONTAINED ELEPHANTS.
SOME OF THE CAGES CONTAINED SOME ELEPHANTS WHICH WERE AMONG THE ANIMALS.

[5]. ONE CONTAINED A TIGER.
ONE OF THE CAGES CONTAINED A TIGER WHICH WAS AMONG THE ANIMALS.

[6]. THE MAN JUMPED FROM THE LORRY.
A MAN JUMPED FROM THE LORRY.

[7]. HE HELD THE DOOR.
THE MAN HELD THE LORRY'S DOOR.

[8]. A MONKEY CAME OUT.
A MONKEY CAME FORTH.

[9]. IT HAD A SAD FACE.
THE MONKEY POSSESSED A SAD FACE.

[10]. THE GIRAFFE ASKED HIM THE REASON FOR THIS.
FROM THE MONKEY, A GIRAFFE INQUIRED THE REASON THAT MONKEY POSSESSED THE FACE.

[11]. HE SAID THAT HE HAD LEFT HIS FOOD AT HOME.
THE MONKEY SAID THAT AT HOME, THAT MONKEY HAD DEPOSITED THAT MONKEY'S FOOD.

[12]. HE WAS HUNGRY.
THE MONKEY WAS HUNGRY.

[13]. A TREE WAS OUTSIDE THE ZOO.
A TREE WAS AT THE ZOO.

[14]. THE GIRAFFE OFFERED HELP.
THE GIRAFFE OFFERED HELP.

[15]. HE GRASPED THE RIPE LEAVES WITH HIS MOUTH.
WITH THE MOUTH WHICH WAS NEAR THE GIRAFFE, THAT GIRAFFE GRASPED SOME RIPE LEAVES.

[16]. HE PULLED THE TREE OUTOF THE GROUND.
FROM THE GROUND, THE GIRAFFE PULLED THE TREE.

[17]. THE TREE FELL ON THE TIGER.
THE TREE FELL NEAR THE TIGER.

[18]. THE TIGER FELL ON THE ELEPHANT.

THE TIGER FELL NEAR THE ELEPHANT WHICH WAS AMONG THE ANIMALS.

[19]. THE ELEPHANT FELL ON THE HORSE.
THE ELEPHANT WHICH THE TIGER FELL NEAR FELL NEAR THE HORSE WHICH WAS AMONG THE ANIMALS.

[20]. THE ZOOKEEPER HEARD THE DISTURBANCE.
A ZOOKEEPER HEARD THE ELEPHANT WHICH THE TIGER FELL NEAR FALL NEAR THE HORSE.

[21]. HE WAS ANGRY.
THE ZOOKEEPER WAS ANGRY.

[22]. THE MONKEY SAID THAT HE DID NOT EAT LEAVES.
THE MONKEY SAID THAT THAT MONKEY DID NOT DEVOUR ANY LEAVES.

[23]. HE ATE BANANAS.
THE MONKEY DEVOURED SOME BANANAS.

Although the following story is shorter than the last one, the influence of global focus is stronger; thus SPAR inevitably makes some errors. The bizarre interpretation of [18] is however due to Sidner-based AR rules predicting both Joan and the attendant for "she" (an agent pronoun) but only Joan for "her" (a normal pronoun); inference is therefore not invoked.

B8 [1]. JOAN PLANNED TO DRIVE TO CAMBRIDGE.
JOAN PLANNED TO DRIVE TO CAMBRIDGE.

[2]. SHE GOT UP AT DAWN.
JOAN GOT UP AT DAWN.

[3]. SHE GAVE HER FRIEND THE KEY OF HER HOUSE.
TO JOAN'S FRIEND, JOAN PRESENTED THE SECURITY KEY WHICH WAS AT JOAN'S HOUSE.

[4]. SHE PUT THE CAT IN THE CELLAR.
NEAR THE HOUSE'S CELLAR, JOAN PUT A CAT.

[5]. SHE GAVE HER FRIEND SOME FOOD FOR IT.
TO JOAN'S FRIEND, JOAN PRESENTED SOME FOOD WHICH WAS FOR THE CAT.

[6]. SHE TOLD HER THAT SHE WOULD RETURN ON FRIDAY.
JOAN INFORMED JOAN'S FRIEND THAT JOAN WOULD RETURN ON FRIDAY.

[7]. SHE WROTE A NOTE FOR THE MILKMAN.
TO A MILKMAN, JOAN WROTE A NOTE.

[8]. SHE PUT OIL IN THE ENGINE OF THE CAR.
NEAR THE ENGINE WHICH WAS NEAR A CAR, JOAN PUT SOME OIL.

[9]. SHE PUT HER CASE IN THE CAR.
NEAR THE CAR, JOAN PUT JOAN'S CASE.

[10]. SHE PUT A BOX OF APPLES IN THE CAR FOR HER SON.

FOR JOAN'S SON, AND NEAR THE CAR, JOAN PUT A BOX WHICH HAD SOME APPLES.

[11]. HE LIKED APPLES.
THE SON LIKED APPLES.

[12]. SHE PUT A CAKE IN THE CAR FOR HER SON'S WIFE.
FOR JOAN'S SON'S WIFE, AND NEAR THE CAR, JOAN PUT A CAKE.

[13]. SHE LIKED CAKE.
JOAN LIKED CAKES.

[14]. SHE PUT A PHOTO IN THE CAR FOR HER MOTHER.
FOR JOAN'S MOTHER, AND NEAR THE CAR, JOAN PUT A PHOTO.

[15]. SHE LIKED PHOTOS.
JOAN LIKED PHOTOS.

[16]. SHE DROVE TO THE GARAGE AT WHICH SHE BOUGHT SOME PETROL.
JOAN DROVE TO THE VEHICLE REPAIR CENTRE WHICH JOAN PURCHASED SOME PETROL AT.

[17]. SHE ASKED THE ATTENDANT TO PUT AIR IN THE TYRES.
JOAN REQUESTED THE ATTENDANT WHO WORKED AT THE VEHICLE REPAIR CENTRE TO PUT SOME AIR NEAR SOME TYRES.

[18]. SHE PAID HER.
THE ATTENDANT PAID JOAN.

[19]. SHE DROVE AWAY.
JOAN DROVE.

The next example uses many of the same concepts as B1. Note that the errors in [3] ("his friend") and [15] ("he") do not propagate to later sentences.

The use of the modifier "which Sam would loan John" in [8] onwards, rather than the more precise "which John asked Sam to loan John", is due to the generator's inability to express complex relative clauses (see footnote in 9.3.1).

<u>B9</u> [1]. JOHN WANTED BILL TO MEND HIS CAR.
JOHN WANTED BILL TO REPAIR JOHN'S CAR.

[2]. HE TOOK IT TO BILL'S GARAGE.
TO BILL'S GARAGE, JOHN CONVEYED THE CAR.

[3]. BILL ASKED HIS FRIEND TO LEND HIM SOME TOOLS.
BILL REQUESTED BILL'S FRIEND TO LOAN BILL SOME REPAIR IMPLEMENTS.

[4]. JOHN SAID THAT HE THOUGHT THAT BILL HAD TOOLS.
JOHN SAID THAT JOHN BELIEVED THAT BILL POSSESSED SOME REPAIR IMPLEMENTS.

[5]. HE PROMISED TO LEND HIM SOME.
JOHN PROMISED THAT JOHN WOULD LOAN BILL SOME REPAIR IMPLEMENTS.

[6]. JOHN DID NOT HAVE THE TOOLS.
JOHN DID NOT POSSESS THE REPAIR IMPLEMENTS WHICH JOHN WOULD LOAN BILL.

[7]. HE ASKED SAM TO LEND HIM SOME.
JOHN REQUESTED SAM TO LOAN JOHN SOME REPAIR IMPLEMENTS.

[8]. HE PROMISED TO RETURN THEM BY FRIDAY.
JOHN PROMISED THAT BEFORE FRIDAY, JOHN WOULD GIVE BACK THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[9]. HE GAVE THEM TO BILL.
TO BILL, JOHN PRESENTED THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[10]. BILL MENDED THE CAR.
BILL REPAIRED THE CAR.

[11]. HE GAVE JOHN HIS TOOLS.
TO JOHN, BILL PRESENTED THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[12]. JOHN FORGOT TO RETURN THEM TO SAM.
JOHN FORGOT TO GIVE BACK THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN TO SAM.

[13]. SAM ASKED HIM FOR THEM.
FROM JOHN, SAM DEMANDED THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[14]. SAM SAID THAT HE HAD PROMISED TO RETURN THEM BY FRIDAY.
SAM SAID THAT JOHN HAD PROMISED THAT BEFORE FRIDAY, JOHN WOULD GIVE BACK THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[15]. HE NEEDED THEM.
JOHN NEEDED THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[16]. JOHN REMEMBERED HIS PROMISE.
JOHN REMEMBERED THAT JOHN HAD PROMISED THAT BEFORE FRIDAY, JOHN WOULD GIVE BACK THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[17]. HE GOT THE TOOLS.
JOHN OBTAINED THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

[18]. HE GAVE THEM TO SAM WITH THANKS.
TO SAM, AND WITH THANKS, JOHN PRESENTED THE REPAIR IMPLEMENTS WHICH SAM WOULD LOAN JOHN.

In the following, the misresolution in [4] leads to only John being in focus when the (non-coreferential) pronouns in [5] are resolved. The way such problems could be diagnosed and repaired is discussed in 10.4.2.

<u>B10</u> [1]. JOHN WENT TO A RESTAURANT.
JOHN WENT TO A RESTAURANT.

[2]. HE ASKED FOR A STEAK.

JOHN DEMANDED A STEAK.

[3]. AFTER EATING IT HE TOLD THE WAITER THAT HE HAD NO MONEY.
AFTER JOHN DEVOURED THE STEAK, JOHN INFORMED A WAITER THAT JOHN POSSESSED NO MONEY.

[4]. HE WAS FURIOUS.
JOHN WAS FURIOUS.

[5]. HE MADE HIM WASH DISHES.
IGNORED BECAUSE OF UNRESOLVABLE PRONOUN.

In the next two texts, adapted from Wilks [1975b], Sidner's rules settle firmly on the whisky for "it" in B11, but demand special mode inference for the "it" in B12.

B11 [1]. JOHN DRANK THE WHISKY FROM A GLASS.
FROM A GLASS, JOHN IMBIBED SOME WHISKY.

[2]. IT FELT WARM IN HIS STOMACH.
THE WHISKY SEEMED WARM AT JOHN'S STOMACH.

B12 [1]. JOHN LEFT THE WINDOW.
JOHN DEPARTED FROM A WINDOW.

[2]. HE DRANK THE WINE ON THE TABLE.
JOHN IMBIBED THE WINE WHICH WAS NEAR A DINING TABLE.

[3]. IT WAS GOOD.
THE WINE WAS GOOD.

Texts B13 to B20 are adapted from appendix A of Tait [1982]. They were originally designed to test the script application capabilities of Tait's system; however, if what is required is straight paraphrase rather than the more complex summarising performed by Tait's system, then shallow processing is (mostly) sufficient. (Although performance for full noun phrases is not as good as for pronouns).

Tait's version of B13 ends "...and took it away with her". In my version, the idiomatic "with her", which appears to violate the c-command restriction, was removed, although SPAR could be extended to recognize it. Also, "it" and "away" were transposed because the analyser cannot deal with "she took it away".

Note that "them" in [3] is a descriptional anaphor, not a definite one; and that the resolution of "she" in [3] to Mary and not Susan is accomplished by a weak inference rule (see 8.4.3).

B13 [1]. MARY WANTED TO GIVE SUSAN A PRESENT.
MARY WANTED TO PRESENT A GIFT TO SUSAN.

[2]. SHE THOUGHT THAT SHE WOULD LIKE A COMPUTER.
MARY BELIEVED THAT SUSAN WOULD LIKE A COMPUTER.

[3]. SHE WENT TO THE SHOP WHICH SOLD THEM.
MARY WENT TO THE SHOP WHICH SOLD SOME COMPUTERS.

[4]. THEY LOOKED EXCITING.
THE COMPUTERS WHICH THE SHOP SOLD LOOKED EXCITING.

[5]. SHE WALKED OVER TO ONE.
MARY WALKED TO ONE OF THE COMPUTERS WHICH THE SHOP SOLD.

[6]. SHE TRIED IT BY WRITING A LITTLE PROGRAM.
BY WRITING A LITTLE PROGRAM WHICH THE COMPUTERS WHICH THE SHOP SOLD OBEYED, MARY TESTED THE COMPUTER WHICH MARY WALKED TO.

[7]. SHE ENJOYED IT.
MARY ENJOYED TESTING THE COMPUTER WHICH MARY WALKED TO BY WRITING THE PROGRAM.

[8]. SHE DECIDED THAT IT WOULD BE A GOOD PRESENT FOR SUSAN.
MARY CONCLUDED THAT THE COMPUTER WHICH MARY WALKED TO WOULD BE A GOOD GIFT WHICH WAS FOR SUSAN.

[9]. SHE PAID FOR IT.
MARY PAID FOR THE COMPUTER WHICH MARY WALKED TO.

[10]. SHE TOOK AWAY IT.
MARY TOOK AWAY THE COMPUTER WHICH MARY WALKED TO.

<u>B14</u> [1]. JOHN WAS HUNGRY.
JOHN WAS HUNGRY.

[2]. HE SENT MARY TO THE KITCHEN.
TO A KITCHEN, JOHN SENT MARY.

[3]. SHE GOT SOME STEAK FROM THE FRIDGE.
FROM A FRIDGE, MARY OBTAINED SOME STEAK.

[4]. SHE MADE A PIE.
MARY ASSEMBLED A PIE WHICH WAS AMONG THE FOOD WHICH WAS NEAR THE FRIDGE.

[5]. JOHN LOVED IT.
JOHN LOVED THE PIE.

<u>B15</u> [1]. JOHN PICKED A CAN OF TUNA OFF THE SHELF.
FROM A SHELF, JOHN SELECTED A CAN WHICH HAD TUNA.

[2]. HE PUT IT IN HIS BASKET.
NEAR JOHN'S BASKET, JOHN PUT THE CAN.

[3]. HE PAID FOR IT.
JOHN PAID FOR THE CAN.

[4]. HE WENT HOME.
JOHN WENT HOME.

Errors in processing can sometimes lead to bizarre but essentially coherent interpretations of a story. An example is SPAR's failure to recognize that the "present" is the "zebra" in [4] below. More sophisticated inference would be needed to verify that this identity does not break the information constraint. However, the *knowledge* needed for such verification is in fact already present in the representation.

B16 [1]. SPUD WENT TO A SHOP.
SPUD WENT TO A SHOP.

[2]. HE DECIDED THAT A ZEBRA WOULD BE A GOOD PRESENT FOR MURIEL.
SPUD CONCLUDED THAT A ZEBRA WOULD BE A GOOD GIFT WHICH WAS FOR MURIEL.

[3]. HE BOUGHT ONE.
SPUD BOUGHT A ZEBRA.

[4]. HE GAVE HER HIS PRESENT.
TO THE ZEBRA WHICH SPUD BOUGHT, SPUD PRESENTED SPUD'S GIFT.

[5]. SHE LOVED HIM.
THE ZEBRA WHICH SPUD BOUGHT LOVED SPUD.

The wrong interpretation in [7] below (and in [4] above) could perhaps be avoided by depreferring an (initial) "he" or "she" reference to an animal if a human referent is available. However this error does not propagate to [8].

B17 [1]. JOHN WENT TO THE ZOO.
JOHN WENT TO A ZOO.

[2]. HE SAW THE LIONS.
JOHN SAW THE LIONS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO.

[3]. HE WANTED TO GIVE MURIEL A PRESENT.
JOHN WANTED TO PRESENT A GIFT TO MURIEL.

[4]. HE THOUGHT THAT SHE WOULD LIKE A TIGER.
JOHN BELIEVED THAT MURIEL WOULD LIKE A TIGER.

[5]. HE WENT TO A SHOP WHICH SOLD THEM.
JOHN WENT TO A SHOP AND THE SHOP SOLD SOME TIGERS.

[6]. HE GOT ONE.
JOHN ACQUIRED ONE OF THE TIGERS WHICH THE SHOP SOLD.

[7]. SHE LOVED IT.
THE TIGER WHICH JOHN ACQUIRED LOVED JOHN ACQUIRING THAT TIGER.

[8]. IT ATE HER.
THE TIGER WHICH JOHN ACQUIRED DEVOURED MURIEL.

SPAR's processing of the next three stories exhibits few phenomena not seen in the examples so far. However, the AR rules are able to deal correctly with a number of potentially difficult "they"s. Note in particular [4] in B18, [6] in B19, and [4] in B20, where established referent sets not mentioned in the previous sentence are correctly preferred over new sets constructed from entities mentioned in the previous sentence.

B18 [1]. MIKE WENT TO THE ZOO WITH MARY.
WITH MARY, MIKE WENT TO A ZOO.

[2]. THEY SAW THE ZEBRAS.
MARY AND MIKE SAW THE ZEBRAS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO.

[3]. MIKE FED THE ELEPHANT.
MIKE FED THE ELEPHANT WHICH WAS AMONG THE ANIMALS.

[4]. THEY SAW THE MONKEYS.
MARY AND MIKE SAW THE MONKEYS WHICH WERE AMONG THE ANIMALS.

[5]. THEY WENT TO A RESTAURANT.
MARY AND MIKE WENT TO A RESTAURANT.

[6]. THEY ASKED FOR STEAKS.
MARY AND MIKE DEMANDED SOME STEAKS.

[7]. THEY ATE THEM.
MARY AND MIKE DEVOURED THE STEAKS.

[8]. THEY TOLD THE WAITER THAT THEY HAD NO MONEY.
MARY AND MIKE INFORMED A WAITER THAT MARY AND MIKE POSSESSED NO MONEY.

[9]. THE WAITER WAS FURIOUS.
THE WAITER WAS FURIOUS.

[10]. THEY RAN AWAY.
MARY AND MIKE FLED.

B19 [1]. MARY WENT TO THE ZOO WITH JOHN.
WITH JOHN, MARY WENT TO A ZOO.

[2]. THEY FED THE MONKEYS SOME PEANUTS.
JOHN AND MARY FED THE MONKEYS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO SOME PEANUTS.

[3]. THEY SAW THE ZEBRA.
JOHN AND MARY SAW THE ZEBRA WHICH WAS AMONG THE ANIMALS.

[4]. THEY WENT TO A RESTAURANT.
JOHN AND MARY WENT TO A RESTAURANT.

[5]. JOHN TOLD THE WAITER THAT HE WANTED A STEAK.
JOHN INFORMED THE WAITER WHO WORKED AT THE RESTAURANT THAT JOHN WANTED TO HAVE A STEAK.

[6]. THEY DID NOT HAVE ANY MONEY.
JOHN AND MARY DID NOT POSSESS ANY MONEY.

[7]. THE WAITER WAS ANGRY.

THE WAITER WAS ANGRY.

[8]. THEY RAN AWAY.
JOHN AND MARY FLED.

B20 [1]. MARY WENT TO THE ZOO WITH JOHN.
WITH JOHN, MARY WENT TO A ZOO.

[2]. THEY SAW THE MONKEYS.
JOHN AND MARY SAW THE MONKEYS WHICH WERE AMONG THE ANIMALS WHICH WERE AT THE ZOO.

[3]. JOHN FED THE ELEPHANT A BANANA.
JOHN FED THE ELEPHANT WHICH WAS AMONG THE ANIMALS A BANANA.

[4]. THEY WENT HOME.
JOHN AND MARY WENT HOME.

Our final three texts show something of SPAR's flexibility, since they represent different domains from the one SPAR is designed for. B21 is from Alshawi [1983]; B22, whose main focus of interest is the Bach-Peters sentence [7], is from Weyters [1985]. B23 is equivalent to B22 as far as anaphor resolution is concerned, but the names given to the characters enable the generator to express the resolution decisions more clearly.

B21 [1]. THE CONDITION OF P971 IS GOOD.
P971'S CONDITION IS GOOD.

[2]. HADDON COLLECTED IT.
HADDON COLLECTED P971.

[3]. HE WAS A BRITISH ACADEMIC.
HADDON WAS A BRITISH ACADEMIC.

[4]. BEVAN WAS A BRITISH MUSEUM-KEEPER.
BEVAN WAS A BRITISH MUSEUM-KEEPER.

[5]. HE COLLECTED P956.
BEVAN COLLECTED P956.

[6]. THE CONDITION OF THIS ONE IS FAIR.
P956'S CONDITION IS FAIR.

[7]. BOTH ARTIFACTS ARE SPEARS.
P956 AND P971 ARE SOME SPEARS.

[8]. SMITH WHO WAS A GERMAN TRADER COLLECTED P316 WHICH IS AN ARROW.
SMITH WAS A GERMAN TRADER AND P316 IS AN ARROW AND SMITH COLLECTED P316.

[9]. HE COLLECTED P612 FROM WOODLARK.
FROM WOODLARK, SMITH COLLECTED P612.

[10]. THIS ARTIFACT IS A NECKLACE.
P612 IS A NECKLACE.

[11]. THE THREE WEAPONS COME FROM DAUI.
P316, P956 AND P971 COME FROM DAUI.

[12]. THE CONDITION OF THE ARROW IS GOOD.
P316'S CONDITION IS GOOD.

[13]. THE ORNAMENT IS POOR.
P612 IS POOR.

B22 [1]. THERE WAS A MAN.
A MAN EXISTED.

[2]. THERE WAS A WOMAN.
A WOMAN EXISTED.

[3]. THE WOMAN LOVED THE MAN.
THE WOMAN LOVED THE MAN.

[4]. THERE WAS ANOTHER WOMAN.
ANOTHER WOMAN EXISTED.

[5]. THE MAN HATED THE WOMAN WHO LOVED HIM.
THE MAN HATED THE WOMAN WHO LOVED THAT MAN.

[6]. THERE WAS ANOTHER MAN.
ANOTHER MAN EXISTED.

[7]. THE WOMAN WHO LOVED HIM MARRIED THE MAN WHO HATED HER.
THE WOMAN WHO LOVED THE MAN WHO HATED THAT WOMAN MARRIED THAT MAN.

B23 [1]. ARTHUR WAS A MAN.
ARTHUR WAS A MAN.

[2]. BARBARA WAS A WOMAN.
BARBARA WAS A WOMAN.

[3]. THE WOMAN LOVED THE MAN.
BARBARA LOVED ARTHUR.

[4]. CAROL WAS ANOTHER WOMAN.
CAROL WAS ANOTHER WOMAN.

[5]. THE MAN HATED THE WOMAN WHO LOVED HIM.
ARTHUR HATED BARBARA.

[6]. DAVID WAS ANOTHER MAN.
DAVID WAS ANOTHER MAN.

[7]. THE WOMAN WHO LOVED HIM MARRIED THE MAN WHO HATED HER.
BARBARA MARRIED ARTHUR.

Appendix C: The inference component in action

All the text in this appendix in `This Font` was generated directly by SPAR. For the sake of readability, some output has been deleted, and some of what remains has been repositioned on the page.

For each sentence in each example, the output consists of:

(1) The input sentence, with TMN element names shown in square brackets.
(2) The results of P.I. rule (=AR rule) application.
(3) The chains returned by the inference component.

The contributions of other knowledge sources (e.g. configurational constraints) are also shown where relevant.

Example 1: sentences 2-3 of text B3

(Sentence 1 was "Bill went to the zoo with Jill").

```
[2] THEY [THEY1-1] GAVE [GIVE1-1] THE MONKEYS [MONKEY1-1] SOME BANANAS
    [BANANA1-1] WHICH THEY [THEY1-2] ATE [EAT1-7]

After P.I. rule application, options are:

THEY1-1 =        BILL AND JILL
THEY1-2 =        SOME MONKEYS or BILL AND JILL

Returning these chains:

Chain number 1:
THEY1-2 = SOME MONKEYS from positive chain derived as follows:

Candidate starting point:      ref(THEY1-1) GAVE SOME BANANAS TO SOME MONKEYS
Extraction gives:              SOME MONKEYS HAD SOME BANANAS

Anaphor starting point:        ref(THEY1-2) ATE SOME BANANAS
Extraction gives:              ref(THEY1-2) HAD SOME BANANAS

Matching gives THEY1-2 = SOME MONKEYS

[3] THEY [THEY1-1] WENT [GO1-7] TO THE RESTAURANT [RESTAURANT1-4]

After P.I. rule application, options are:

THEY1-1 =        BILL AND JILL or THE MONKEYS WHICH WERE AMONG SOME ANIMALS

Returning these chains:

Chain number 1:
THEY1-1 must be a MAN from positive chain derived as follows:

Anaphor starting point:        ref(THEY1-1) WENT TO A RESTAURANT
CSI rule DOWANT gives:         ref(THEY1-1) WANTED TO GO TO A RESTAURANT
Extraction gives:              ref(THEY1-1) WANTED TO BE AT A RESTAURANT
Object-related inference on RESTAURANT1 gives:
```

ref(THEY1-1) WANTED TO EAT AT A RESTAURANT

Matching with no-exit rule WANTEAT gives:

THEY1-1 must be MAN

For THEY1-1: preferring BILL AND JILL over THE MONKEYS WHICH WERE AMONG SOME ANIMALS

Example 2: sentences 1-3 of text B1

[1] JOHN [JOHN1-6] PROMISED [PROMISE1-43] BILL [BILL1-3] THAT HE [HE1-1] WOULD MEND [MEND1-8] HIS [HIS1-1] CAR [CAR1-2]

After P.I. rule application, options are:

HE1-1 = JOHN or BILL
HIS1-1 = JOHN or BILL

Returning these chains:

Chain number 1:
HE1-1 = JOHN from positive chain derived as follows:

Anaphor starting point: JOHN PROMISED BILL THAT ref(HE1-1) WOULD MEND ref(HIS1-1)'S CAR

Matching with no-exit rule FFPROM gives:

HE1-1 = JOHN

Chain number 2:
HIS1-1 = BILL from positive chain derived as follows:

Anaphor starting point: JOHN PROMISED BILL THAT ref(HE1-1) WOULD MEND ref(HIS1-1)'S CAR

CSI rule PROMB gives: BILL WANTED ref(HE1-1) TO MEND ref(HIS1-1)'S CAR

Extraction gives: BILL WANTED ref(HIS1-1)'S CAR TO FUNCTION

Matching with no-exit rule WP gives:

HIS1-1 = BILL

[2] HE [HE1-1] TOOK [TAKE5-1] IT [ITO-1] TO HIS [HIS1-1] FRIEND [FRIEND1-1] POSS GARAGE [GARAGE1-1]

After P.I. rule application, options are:

HE1-1 = JOHN or BILL
ITO-1 = BILL'S CAR
HIS1-1 = JOHN

Returning these chains:

Chain number 1:
ITO-1 = BILL'S CAR and HE1-1 = JOHN from positive chain derived as follows:

Candidate starting point: JOHN PROMISED BILL THAT JOHN WOULD MEND BILL'S CAR

CSI rule FFPROM gives: JOHN WANTED TO MEND BILL'S CAR

Anaphor starting point: ref(HE1-1) TOOK ref(IT0-1) TO ref(HIS1-1)'S FRIEND'S GARAGE

Extraction gives: ref(HE1-1) WANTED ref(IT0-1) TO BE AT ref(HIS1-1)'S FRIEND'S GARAGE

Object-related inference on GARAGE1 gives: ref(HE1-1) WANTED PERSONS TO MEND ref(IT0-1) AT ref(HIS1-1)'S FRIEND'S GARAGE

Matching gives IT0-1 = BILL'S CAR and HE1-1 = JOHN

Chain number 2:
IT0-1 = BILL'S CAR and HE1-1 = BILL from positive chain derived as follows:

Candidate starting point: JOHN PROMISED BILL THAT JOHN WOULD MEND BILL'S CAR

CSI rule PROMB gives: BILL WANTED JOHN TO MEND BILL'S CAR

Anaphor starting point: ref(HE1-1) TOOK ref(IT0-1) TO ref(HIS1-1)'S FRIEND'S GARAGE

Extraction gives: ref(HE1-1) WANTED ref(IT0-1) TO BE AT ref(HIS1-1)'S FRIEND'S GARAGE

Object-related inference on GARAGE1 gives: ref(HE1-1) WANTED PERSONS TO MEND ref(IT0-1) AT ref(HIS1-1)'S FRIEND'S GARAGE

Matching gives IT0-1 = BILL'S CAR and HE1-1 = BILL

Applying CCP criterion
For HE1-1: preferring JOHN over BILL

[3] HE [HE1-2] TRIED [TRY1-1] TO PERSUADE [PERSUADE1-1] HIS [HIS1-1] FRIEND [FRIEND1-5] THAT HE [HE1-3] SHOULD LEND [LEND1-1] HIM [HIM1-1] SOME TOOLS [TOOL1-1]

After P.I. rule application, options are:

HE1-2 = JOHN or JOHN'S FRIEND
HIS1-1 = JOHN
HE1-3 = JOHN or JOHN'S FRIEND
HIM1-1 = JOHN'S FRIEND or JOHN

After applying configurational constraints, options are:

HE1-2 = JOHN
HIS1-1 = JOHN
HE1-3 = JOHN or JOHN'S FRIEND
HIM1-1 = JOHN'S FRIEND or JOHN

Returning these chains:

Chain number 1:
HIM1-1 = ref(HE1-2) from positive chain derived as follows:

Anaphor starting point: ref(HE1-2) TRIED TO PERSUADE ref(HIS1-1)'S

FRIEND THAT ref(HE1-3) SHOULDED LEND
ref(HIM1-1) SOME TOOLS

Extraction gives: ref(HE1-2) WANTED ref(HIM1-1) TO HAVE SOME TOOLS

Matching with no-exit rule WANTHAVE gives:
HIM1-1 = ref(HE1-2)

For HE1-3: preferring JOHN'S FRIEND over JOHN
For HIM1-1: preferring JOHN over JOHN'S FRIEND

Example 3: sentences 2-3 of text B13

(Sentence 1 was "Mary wanted to give Susan a present").

[2] SHE [SHE1-1] THOUGHT [THINK1-13] THAT SHE [SHE1-2] WOULD LIKE [LIKE1-1] A COMPUTER [COMPUTER1-1]

After P.I. rule application, options are:

SHE1-1 = MARY or SUSAN
SHE1-2 = MARY or SUSAN

Returning these chains:

Chain number 1:
SHE1-1 = MARY and SHE1-2 = SUSAN from positive chain derived as follows:

Candidate starting point: MARY WANTED TO GIVE A PRESENT TO SUSAN

Anaphor starting point: ref(SHE1-1) THOUGHT THAT ref(SHE1-2) WOULD LIKE A COMPUTER

CSI rule GIVETOPLEASE gives: ref(SHE1-1) WANTED TO GIVE A COMPUTER TO ref(SHE1-2)

Matching gives SHE1-1 = MARY and SHE1-2 = SUSAN

[3] SHE [SHE1-3] WENT [GO1-5] TO THE SHOP [SHOP2-4] WHICH SOLD [SELL1-2] THEM [THEM2-1]

After P.I. rule application, options are:

SHE1-3 = MARY or SUSAN

Returning these chains:

(Chains 1 and 2 arise from object-oriented inferences on SHOP2, *and merely verify (unhelpfully but correctly) that* SHE1-3 *must be a* MAN *and not a* BEAST.*)*

Chain number 3:
SHE1-3 = MARY from positive chain derived as follows:
Weak CSI matching using rule ON-STAGE-RULE gives:
SHE1-3 = MARY

Example 4: sentence 2 of text A31

(Sentence 1 was "The soldiers fired at the women").

[2] THEY [THEY1-1] DID NOT FALL [FALL1-5]

After P.I. rule application, options are:

THEY1-1 = SOME SOLDIERS or SOME WOMEN

Returning these chains:
THEY1-1 = SOME WOMEN from positive chain derived as follows:

Candidate starting point:	SOME SOLDIERS FIRED AT SOME WOMEN
CSI rule FFGOAL gives:	A THING STRIKKED *(i.e. struck!)* THE WOMEN WHO THAT THING MOVED TO
CSI rule STRIKHURT gives:	SOME WOMEN WERE NOT WHOLE
CSI rule HURTFALL gives:	SOME WOMEN FELL
Anaphor starting point:	ref(THEY1-1) DID NOT FALL

Matching gives THEY1-1 = SOME WOMEN

Appendix D: SPAR's CSI rules.

This appendix contains SPAR's complete database of CSI rules. See 8.2.1 for details of the interpretation of the various fields.

```
(ASKTELL ((MOTIVATES A TE <>))
   (A ASK (FOR S))
   (S *HUM)
   (TE TELL (SUBJ S)))

(CATCHHURT ((RESULT C H >))
   (C CATCH1 (OBJE A))
   (A *HUM)
   (H BE (SUBJ A) (STATE WHOLE) (NEG NEG)))

(DOWANT ((MOTIVATES W D <>))
   (W WANT (SUBJ A) (OBJE D))
   (A MAN)
   (D ACT (SUBJ A)))

(FFGOAL ((RESULT D G >))
   (D *DO (GOAL G))
   (G *DO))

(FFPROM ((NO-EXIT P NIL) (RESULT P W >))
   (P PROMISE1 (SUBJ S) (OBJE D))
   (S *HUM)
   (D ACT (SUBJ S))
   (W WANT (SUBJ S) (OBJE D)))

(GIVETOPLEASE ((SUGGESTS H W <>))
   (H THINK (SUBJ A) (OBJE L))
   (L LIKE1 (SUBJ C) (FOR B))
   (W WANT (SUBJ A) (OBJE G))
   (G GIVE1 (SUBJ A) (OBJE B) (FOR C))
   (A *HUM)
   (B *PHYSOB)
   (C *HUM))

(HAVUSE ((ENABLES H U >))
   (H HAVE (SUBJ P) (OBJE Q))
   (U USE (SUBJ P) (OBJE Q))
   (P *HUM)
   (Q *INAN))

(HURTFALL ((RESULT H F <>))
   (H BE (SUBJ P) (STATE WHOLE) (NEG NEG))
   (P *PHYSOB)
   (F FALL1 (SUBJ P)))

(HURTSQUEAL ((MOTIVATES H S <))
   (H BE (SUBJ A) (STATE WHOLE) (NEG NEG))
   (A *HUM)
   (S SQUEAL1 (SUBJ A)))
```

```
(JUDGEGIFT ((RESULT G J >))
   (G GIVE (OBJE A) (FOR B))
   (J FEEL (SUBJ B) (FOR A))
   (A *PHYSOB)
   (B *HUM))

(JUDGFOOD ((RESULT C F <>))
   (C CONSUME1 (SUBJ H) (FOR X))
   (H *HUM)
   (X *INAN)
   (F FEEL (SUBJ H) (FOR X)))

(LIKEGOOD ((SUGGESTS F G) (SUGGESTS G F))
   (F FEEL (SUBJ H) (FOR X))
   (H *HUM)
   (X *ENT)
   (G BE (SUBJ X) (STATE GOOD)))

(PERSB ((SUGGESTS PE W <>))
   (PE PERSUADE1 (SUBJ H) (OBJE D))
   (H *HUM)
   (D *DO)
   (W WANT (SUBJ H) (OBJE D)))

(PROMB ((SUGGESTS PR W <>))
   (PR PROMISE1 (OBJE A) (FOR H))
   (A *DO)
   (H *HUM)
   (W WANT (SUBJ H) (OBJE A)))

(STRIKHURT ((RESULT S H <>))
   (S STRIK (SUBJ P) (OBJE Q))
   (P *PHYSOB)
   (Q *PHYSOB)
   (H BE (SUBJ Q) (STATE WHOLE) (NEG NEG)))

(WANTEAT ((NO-EXIT W NIL))
   (W WANT (SUBJ S) (OBJE E))
   (S *HUM)
   (E CONSUME1 (SUBJ S)))

(WANTHAVE ((NO-EXIT W NIL))
   (W WANT (SUBJ S) (OBJE H))
   (S *HUM)
   (H HAVE (SUBJ S) (OBJE P))
   (P *PHYSOB))

(WANTPAY ((MOTIVATES W P <>))
   (W WANT (SUBJ A) (OBJE D))
   (D ACT (SUBJ B))
   (A *HUM)
   (B *HUM)
   (P PAY1 (SUBJ A) (FOR B)))

(WP ((SUGGESTS WP W <>)
```

```
(NO-EXIT W NIL))
(WP FUNC (SUBJ S))
(S *PHYSOB (POSS H))
(H *HUM)
(W WANT (SUBJ H) (OBJE WP)))
```

References

Alshawi, H. (1983) *Memory and Context Mechanisms for Automatic Text Processing*, University of Cambridge Computer Laboratory TR-60

Alshawi, H. (1985) *Processing Dictionary Definitions with Phrasal Pattern Hierarchies*, University of Cambridge Computer Laboratory

Anderson, S.R. (1977) 'Comments on the paper by Wasow' in Culicover, Wasow and Akmajian (eds.), *Formal Syntax*, Academic Press, pp.361-377

Boguraev, B.K. (1979) *Automatic Resolution of Linguistic Ambiguities*, University of Cambridge Computer Laboratory TR-11

Bolinger, D. (1979) 'Pronouns in Discourse' in Givon, T. (ed.), *Syntax and Semantics, Vol. 12: Discourse and Syntax*, Academic Press, pp.289-309

Bosch, P. (1980) 'The Modes of Pronominal Reference and their Constraints' in Kreiman and Ojeda (eds.), *Papers from the Parasession on Pronouns and Anaphora*, University of Chicago, Chicago, Illinois, pp.64-78

Bundy, A. et al. (1979) 'Solving Mechanics Problems using Meta-level Inference', *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, Tokyo, Japan, pp1017-1027

Cantrall, W.R. (1975) 'Favored Structures and Intonational Limitations' in Grossman, San and Vance (eds.), *Papers from the Parasession on Functionalism*, University of Chicago, Chicago, Illinois, pp.36-46

Carter, D.M. (1984) 'On the Fass and Wilks Proposal to Use "Polysemy Rules"', *Computational Linguistics, vol. 10(2)*, 147

Cater, A.W.S. (1981) *Analysis and Inference for English*, University of Cambridge Computer Laboratory TR-19

Charniak, E. (1972) *Towards a model of children's story comprehension*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology AI-TR-266

Charniak, E. (1978) 'On the use of Framed Knowledge in Language Comprehension', *Artificial Intelligence, vol. 11*, 225-265

Charniak, E. (1981a) 'The Case-slot Identity Theory', *Cognitive Science, vol. 5(3)*, 285-292

Charniak, E. (1981b) 'A Common Representation for Problem-solving and Language Comprehension Information', *Artifical Intelligence, vol. 16*, 225-255

Chomsky, N. (1971) 'Deep Structure, Surface Structure and Semantic Interpretation' in Steinberg and Jakobovitz (eds.), *Semantics: an Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, Cambridge University Press

Cooper, R. (1979) 'The Interpretation of Pronouns' in Heny and Schnelle (eds.), *Syntax and Semantics, Vol. 10*, Academic Press, pp.61-92

Crain, S. and Steedman, M. (1985) 'Context and the psychological syntax processor' in Dowty, Karttunen and Zwicky (eds.), *Natural language parsing*, Cambridge University Press, pp.320-358

Cullingford, R.E. (1978) *Script application: computer understanding of newspaper stories*, Department of Computer Science, Yale University, research report 116

Cullingford, R.E. (1981) 'SAM' in Schank, R.C. and Riesbeck, C.K. (eds.), *Inside Computer Understanding*, Lawrence Earlbaum Associates, pp.75-119

Dale, R. (1985) 'The Use of Focus in the Computer Generation of Referring Expressions', *Abstracts for the Third International Colloquium on the Interdisciplinary Study of Natural Language, Kleve, Niederrhein, West Germany*

Danlos, L. (1984) 'An Algorithm for Automatic Generation', *Proceedings of the Sixth European Conference on Artificial Intelligence*, Pisa, Italy, pp213-216

DeJong, G.M. (1979) *Skimming newspaper stories by computer*, Department of Computer Science, Yale University, research report 104

Dyer, M.G. (1982) *In-depth Understanding*, Department of Computer Science, Yale University, research report 219

van Eijck, J. (1983) 'Discourse representation theory and plurality' in ter Meulen, A. (ed.), *Studies in model theoretic semantics*, Groningen: Amsterdam, pp.85-106

Evans, G. (1980) 'Pronouns', *Linguistic Inquiry, vol. 11*, 337-362

Fahlman, S.E. (1979) *NETL: A System for Representing and Using Real-World Knowledge*, M.I.T. Press, Cambridge, MA

Fass, D.C. and Wilks, Y.A. (1983) 'Preference Semantics, Ill-formedness, and Metaphor', *American Journal of Computational Linguistics, vol. 9(3-4)*, 178-187

Frazier, L. and Fodor, J. (1979) 'The Sausage Machine: A New Two-Stage Parsing Model', *Cognition, vol. 6*, 291-325

Goldman, N. (1975) 'Conceptual Generation' in Schank, R.C. (ed.), *Conceptual Information Processing*, North-Holland, pp.289-371

Granger, R.H. (1980) 'When Expectation Fails: Towards a self-correcting inference system', *Proceedings of the First Annual National Conference on Artificial Intelligence*, Stanford University, Stanford, California, pp301-305

Granger, R.H., et al (1983) 'Scruffy Text Understanding: Design and Implementation of the Nomad System', *Proceedings of the ACL Conference on Applied Natural Language Processing*, Santa Monica, California, pp104-106

Grosz, B.J. (1977) *The Representation and Use of Focus in Dialogue Understanding*, Technical Note 151, Artificial Intelligence Center, SRI International, Menlo Park, California

Grosz, B.J. (1978) 'Focusing in Dialog' in Waltz, D. (ed.), *TINLAP-II: Theoretical issues in natural language processing - 2*, University of Illinois at Urbana-Champaign, pp.96-103

Grosz, B.J., Joshi, A.K. and Weinstein, S. (1983) 'Providing a Unified Account of Definite Noun Phrases in Discourse', *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, M.I.T., pp44-50

Gruber, J.S. (1976) *Lexical Structures in Syntax and Semantics*, North-Holland

Guenthner, F. and Lehmann, H. (1983) 'Rules for Pronominalisation', *Proceedings of the First Conference of the European Chapter of the Association for Computational Linguistics*, Pisa, Italy, pp144-151

Halliday, M.A.K. (1967) 'Theme and Information in the English Clause' in Kress, G. (1976) (ed.), *Halliday: System and Function in Language*, Oxford University Press, pp.174-188

Halliday, M.A.K. and Hasan, R. (1976) *Cohesion in English*, Longman

Herskovits, A. (1973) *The Generation of French from a Semantic Representation*, Computer Science Department, Stanford University, Memo AIM-212

Hintikka, J. and Carlson, L. (1979) 'Conditionals, Generic Quantifiers and other applications of subgames' in Saarinen, E. (ed.), *Game-Theoretical Semantics*, Dordrecht: Reidel, pp.179-214

Hirst, G.J. (1983) *Semantic Interpretation Against Ambiguity*, Department of Computer Science, Brown University, Technical Report CS-83-25

Hirst, G.J. (1981) *Anaphora in Natural Language Understanding: a Survey*, Springer-Verlag

Hobbs, J.R. (1976) *Pronoun Resolution*, Department of Computer Sciences, City College, City University of New York, Research Report 76-1

Hobbs, J.R. (1979) 'Coherence and Coreference', *Cognitive Science, vol. 3*, 67-90

Hobbs, J.R. (1983) 'An Improper Treatment of Quantification in Ordinary English', *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, M.I.T., pp57-63

Hobbs, J.R. and Moore, R.C. (eds.) (1985) *Formal Theories of the Commonsense World*, Ablex

Kameyama, M. (1985) *Zero Anaphora: the case of Japanese*, PhD Dissertation, Stanford University Department of Linguistics

Kamp, H. (1981) 'A Theory of Truth and Semantic Representation' in Groenendijk and Janssen (eds.), *Formal Methods in the Study of Language*, Groningen: Amsterdam, pp.1-41

Kantor, R.N. (1977) *The management and comprehension of discourse connection by pronouns in English*, PhD thesis, Department of Linguistics, Ohio State University

Kempson, R.M. (1983) 'Definite NPs and Context-Dependence: a Unified Theory of Anaphora', paper presented at the autumn meeting of the Linguistic Association of Great Britain, University of Newcastle, September 1983.

Klein, E. (1984) 'Verb Phrase Ellipsis in Discourse Representation Theory', unpublished manuscript

Lasnik, H. (1976) 'Remarks on Coreference', *Linguistic Analysis, vol. 2*, 1-22

Lebowitz, M. (1980) *Generalization and Memory in an Integrated Understanding System*, Department of Computer Science, Yale University, Research Report 186

Lees, R.B. and Klima, E.S. (1963) 'Rules for English Pronominalisation', *Language, vol. 39*, 17-28

Lehnert, W.G., Dyer, M.G., Johnson, P.N., Yang, C.J. and Harley, S. (1983) 'BORIS - An Experiment in In-Depth Understanding of Narratives', *Artificial Intelligence, vol. 20(1)*, 15-62

Lockman, A. and Klappholz, A.D. (1980) 'Towards a Procedural Model of Contextual Reference Resolution', *Discourse Processes, vol. 3*, 25-71

Lyons, J. (1968) *Introduction to Theoretical Linguistics*, Cambridge University Press

McDonald, D.D. (1981) *Natural Language Generation as a Computational Problem: an Introduction*, University of Massachusetts at Amherst, COINS Technical Report 81-33

McKeown, K.R. (1982) 'The TEXT system for natural language generation: an overview', *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, pp113-120

McKeown, K.R. (1985) *Text Generation*, Cambridge University Press

Mellish, C.S. (1982) *Incremental Evaluation: An Approach to the Semantic Interpretation of Noun Phrases*, University of Sussex Cognitive Studies Programme

Norvig, P. (1983) 'Frame activated inferences in a story understanding program', *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, pp624-626

Partee, B.H. (1972) 'Opacity, Coreference, and Pronouns' in Harman and Davidson (eds.), *Semantics of Natural Language*, Dordrecht: Reidel, pp.415-441

Partee, B.H. (1978) 'Bound Variables and Other Anaphors' in Waltz, D. (ed.), *TINLAP-II: Theoretical issues in natural language processing - 2*, University of Illinois at Urbana-Champaign, pp.79-85

Pause, P.E. (1984) 'The Interpretation of Anaphoric Expressions in Con3tra', *Proceedings of the Sixth European Conference on Artificial Intelligence*, Pisa, Italy, pp232

Quillian, M.R. (1968) 'Semantic Memory' in Minsky, M.L. (ed.), *Semantic Information Processing*, M.I.T. Press, pp.227-270

Reinhart, T. (1976) *The Syntactic Domain of Anaphora*, PhD Dissertation, M.I.T., Cambridge, Massachusetts

Reinhart, T. (1982) *Pragmatics and Linguistics: an analysis of sentence topics*, Indiana University Linguistics Club

Reinhart, T. (1983) *Anaphora and Semantic Interpretation*, Croom Helm

Rieger, C.J. (1974) *Conceptual Memory: a theory and computer program for processing the meaning content of natural language utterances*, Computer Science Department, Stanford University, AIM-233

Rieger, C.J. (1975) 'Conceptual Memory and Inference' in Schank, R.C. (ed.), *Conceptual Information Processing*, Amsterdam: North-Holland, pp.157-288

Sampson, G. (1983) 'Fallible Rationalism and Machine Translation', *Proceedings of the First Conference of the European Chapter of the Association for Computational Linguistics*, Pisa, Italy, pp86-89

Schank, R.C. (1973) 'Identification of the Conceptualisations Underlying Natural Language' in Schank and Colby (eds.), *Computer Models of Thought and Language*, Freeman, pp.187-247

Schank, R.C. (1975) *Conceptual Information Processing*, Amsterdam: North-Holland

Schank, R.C and Abelson, R.P. (1977) *Scripts, Plans, Goals and Understanding*, Lawrence Earlbaum Associates

Schmolze, J.G. and Brachman, R.J. (eds.) (1982) *Proceedings of the 1981 KL-ONE workshop*, Fairchild Technical Report No. 618

Sidner, C.L. (1979) *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*, M.I.T. Artificial Intelligence Laboratory TR-537

Sidner, C.L. (1979b) 'Disambiguating References and Interpreting Sentence Purpose in Discourse' in Winston and Brown (eds.), *Artificial Intelligence: an M.I.T. perspective (Vol. I)*, M.I.T. Press, Cambridge, MA, pp.231-252

Sidner, C.L. (1981) 'Focusing for Interpretation of Pronouns', *American Journal of Computational Linguistics, vol. 7*, 217-231

Sidner, C.L. (1983) 'Focusing in the Comprehension of Definite Anaphora' in Brady and Berwick (eds.), *Computational Models of Discourse*, M.I.T. Press, Cambridge, MA, pp.267-330

Smith, N. and Wilson, D. (1979) *Modern Linguistics*, Harvester Press

Sparck Jones, K. (ed.) (1984) 'Basic Semantics Information', University of Cambridge Computer Laboratory.

Stenning, K. (1978) 'Anaphora as an Approach to Pragmatics' in Halle, Bresnan and Miller (eds.), *Linguistic Theory and Psychological Reality*, M.I.T. Press, Cambridge, MA, pp.164-200

Tait, J.I. (1982) *Automatic Summarising of English Texts*, University of Cambridge Computer Laboratory Technical Report 47

Tait, J.I. and Sparck Jones, K. (1983) *Automatic Search Term Variant Generation For Document Retrieval*, University of Cambridge Computer Laboratory (British Library R&D Report 5793)

VanLehn, K.A. (1978) *Determining the Scope of English Quantifiers*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AI-TR-483

Webber, B.L. (1978a) *A Formal Approach to Discourse Anaphora*, Garland

Webber, B.L. (1978b) 'Description formation and discourse model synthesis' in Waltz, D. (ed.), *TINLAP-II: Theoretical issues in natural language processing - 2*, University of Illinois at Urbana-Champaign, pp.42-50

Weyters, T. (1985) 'Denotation Resolution Algorithm For Texts (DRAFT)', *Abstracts for the Third International Colloquium on the Interdisciplinary Study of Natural Language, Kleve, Niederrhein, West Germany*

Wiese, B. (1983) 'Anaphora by Pronouns', *Linguistics, vol. 21*, 373-417

Wilensky, R. (1978) *Understanding Goal-based Stories*, Department of Computer Science, Yale University, Research Report 140

Wilensky, R. (1981) 'PAM' in Schank and Riesbeck (eds.), *Inside Computer Understanding*, Lawrence Earlbaum Associates, pp.136-179

Wilensky, R. (1983) *Planning and Understanding*, Addison-Wesley

Wilks, Y.A. (1975a) 'An Intelligent Analyzer and Understander of English', *Communications of the A.C.M., vol. 18*, 264-274

Wilks, Y.A. (1975b) 'A Preferential, Pattern-seeking Semantics for Natural Language Inference', *Artificial Intelligence, vol. 6*, 53-74

Wilks, Y.A. (1975c) 'Preference Semantics' in Keenan, E. (ed.), *The Formal Semantics of Natural Language*, Cambridge University Press, pp.329-348

Wilks, Y.A. (1976) 'Parsing English' in Charniak and Wilks (eds.), *Computational Semantics*, North-Holland, pp.89-100,155-184

Wilks, Y.A. (1977a) 'Good and Bad Arguments about Semantic Primitives', *Communication and Cognition, vol. 10*, 181-221

Wilks, Y.A. (1977b) 'What Sort of Taxonomy of Causation do we need for Natural Language Understanding?', *Cognitive Science, vol. 1*, 235-264

Wilks, Y.A. (1977c) 'Natural Language Understanding Systems within the AI Paradigm: a Survey and some Comparisons' in Zampolli, A. (ed.), *Linguistic Structures Processing*, North-Holland, pp.341-398

Wilks, Y.A. (1978) 'Making Preferences More Active', *Artificial Intelligence, vol. 11*, 197-223

Wilks, Y.A., Huang, X. and Fass, D.C. (1985) 'Syntax, Preference and Right Attachment', *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, California, pp779-784

Winograd, T. (1972) *Understanding Natural Language*, Cambridge University Press

Woods, W. (1972) *The Lunar Sciences Natural Language Information System, Final Report*, Bolt, Beranek and Newman, Cambridge, Massachusetts

Index of terms

accessibility (in discourse representation theory) 28
active anaphors and candidates 146
actor focus (AF) 75,79
actor focus stack (AFS) 75
adequacy (referential) 162
AF (Actor Focus) 75
AFS (Actor Focus Stack) 75
AI (Artificial Intelligence) 6
anaphor 16
anaphor starting point (for inference) 134
anaphor tendril 134
anaphora 16
antecedent 16
AR (Anaphor Resolution) rules 90
ATN (Augmented Transition Network) 40
BA (Boguraev's Analyser) 40
Bach-Peters sentences 26
backward centre 81
basic rule 78
binding (in inference) 134
bound variable pronouns 26
candidate starting point (for inference) 134
candidate tendril 134
cataphora 17
c-command 33
CCP (C-Commanded Pronoun) criterion 109
CD (Conceptual Dependency) 113
centre 81
chain (of inferences) 134
class primitive 37
coherence relation 70
cohesion 16
collective factors 103
collocation 23
common sense inference rule (CSIR) 122,134
conjunction 19
context 56
context of identity (of a TMN node) 58
coreference 17
Corresponds: assertion 53
cospecification 18,19,20,77
CSI (Common Sense Inference) 2,7
CSIR (Common Sense Inference Rule) 122,134
current focus 67
current fragment 56
DE (Discourse Entity) 30
demon 114
demonstrative cospecification 21
denotatum 53
dependent (in formula) 37
descriptional anaphors 21
DF (Discourse Focus) 75
DFS (Discourse Focus Stack) 75

discourse 16
discourse entity (DE) 30
discourse focus (DF) 74,75
discourse focus stack (DFS) 75
discourse representation (DR) 28
discourse representation structure (DRS) 28
discourse representation theory 28
distributive factors 103
donkey sentences 27
DR (Discourse Representation) 28
DRS (Discourse Representation Structure) 28
efficiency (referential) 162
ellipsis 19,21
epithet 23
exophora 17
expected focus 75
explicit TMN node 48
extended mode (in PS) 121
extraction 121
extraction rule 134
FDNP (Full Definite Noun Phrase) 23
focus 67
focus registers 75
focus retention criterion 108
focussing 67
focus update algorithm 76
focus value 67
formula 37
forward centre 81
fragment constructor 56,61
fragmentation 38
full definite noun phrase (FDNP) 23
generic TMN node 58
global focus 68
global inference 112
governor (in formula) 37
habitability 38
hearer 16
heterogeneous inference systems 113,118
ID (Invoking Description) 30
implicit TMN node 48
inference 134
inferred specification 92
information constraint 23
intermediate form (for formulas) 51
inverted nominal formula 37
invoking description (ID) 30
lexical cohesion 19
local focus 68
local inference 112
MGC (Minimal Governing Category) 32
MOP (Memory Organisation Packet) 118
need-driven generation 165
negative matches and chains 139
no-exit inference rules 138

nominal ellipsis 22
nominal substitution 21
non-vacuous binding 137
normal pronouns 79
null TMN node 58
PAF (Potential Actor Focus) 75
paraplate 39
parasitic pronoun 94
paycheck sentences 22
PDF (Potential Discourse Focus) 75
personal cospecification 20
p-focus 67,72
PI (Pronoun Interpretation) rules 76
positive chains 138
potential actor focus (PAF) 75
potential discourse focus (PDF) 75
PP (prepositional phrase) 34
pragmatic pronouns 26
primitive 37
principle of anaphoric success 87
PRL (Possible Referent List) 114
pronoun interpretation (PI) rules 76
pronouns of laziness 22
PS (Preference Semantics system) 36
pseudo-text (PT) 42
PT (Pseudo-Text) 42
recency rule 78
ref and rel 53
reference 17
referential adequacy 162
referential efficiency 162
reiteration 23
repetition criterion 107
R-pronouns 32
script 116
semantic formula 37
semantic primitive 37
sense-oriented representation 44
shallow processing 2
shallow processing hypothesis 3,131
speaker 16
Specialisation: assertion 53
specialised inference systems 113,116
specific TMN node 58
specification 18
starting point (for inference) 134
state of focus 67
status (of a TMN node) 58
strong preference 80
structure matcher 59
substitution 19,21
TAU (Thematic Affect Unit) 118
template 38
tendril 134
text 16

text model 46
text model network (TMN) 48
theme 73
TMN (Text Model Network) 48
topic 72
undetermined anaphor 103
uniform inference systems 113,119
vacuous binding 137
weak inference rule 154
weak preference 80
word sense network (WSN) 47
WSN (Word Sense Network) 47