Evaluating the State of the Art

The FRACAS Consortium

Robin Cooper, Dick Crouch, Jan van Eijck,
Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp,
Manfred Pinkal, Massimo Poesio, Steve Pulman

FraCaS

A Framework for Computational Semantics
LRE 62-051

Deliverable D10

January 3, 1995

LRE 62-051 FRACAS

A Framework for Computational Semantics

CWI Amsterdam

University of Edinburgh
Centre for Cognitive Science and Human Communication Research Centre

Universität des Saarlandes
Department of Computational Linguistics

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung

SRI Cambridge

For copies of reports, updates on project activities and other FRACAS-related information, contact:

The FRACAS Project Administrator
University of Edinburgh
Centre for Cognitive Science
2 Buccleuch Place
Edinburgh EH8 9LW, Scotland, UK
`fracas@cogsci.ed.ac.uk`

Copies of reports and other material can also be accessed via anonymous ftp from ftp.cogsci.ed.ac.uk, directory pub/FRACAS.

Contents

1 Preamble **10**

1.1 Practical applications of semantics 11

1.2 Overview of this document 13

2 The Interface to Syntax **14**

2.1 Introductory remarks 14

2.2 Modular and Integrative Approaches 15

2.3 Modelling Non-local Semantic Phenomena 18

2.4 Formalisms and Processing Models 21

2.5 Modular Systems for Syntactic-Semantic Processing 23

3 Underspecification **26**

3.1 The Case for Underspecification 26

3.2 Background 28

3.2.1 Ambiguity 28

3.2.2 Vagueness, Indefiniteness and Uninformativeness 30

3.3 Approaches to Underspecification 31

3.3.1 Comparing Theories of Underspecification 31

3.3.2 QLF 31

3.3.3 Underspecified DRSs 32

3.3.4 Conversation Representation Theory 35

4 Contextual Reasoning 37

4.1 Techniques 38

4.1.1 Marker passing 38

4.1.2 Deduction 38

4.1.3 Abduction 39

4.1.4 Constraint satisfaction 40

4.1.5 Sortal restrictions 41

4.2 Alternatives 42

4.2.1 Structural preference heuristics 42

4.2.2 Statistical methods 43

5 Inference and Evaluation 44

5.1 Introduction 44

5.2 What types of reasoning are useful? 45

5.2.1 Reasoning about valid consequence 45

5.2.2 Kinds of models, kinds of languages 45

5.2.3 Reasoning about Particular Models 49

5.2.4 Proof Strategies for Valid Inference 50

5.2.5 Reasoning About Plausible Consequence 50

5.2.6 Proof Methods for Plausible Consequence 51

5.3 Inference and Evaluation in Particular Approaches 51

5.3.1 Discourse Representation Theory 51

5.3.2 Dynamic and Update Semantics 52

5.3.3 Monotonic Semantics 52

5.3.4 Property Theory 52

5.3.5 Situation Semantics 53

6 Lexical Semantics 54

6.1 Introduction 54

6.2 Categorematic and Syncategorematic Specifications 54

6.3 Lexical and Computational Semantics 60

7 Survey of Implementations 63

7.1 The Core Language Engine (CLE) 64

7.2 $\mathcal{NLL}$ 65

7.3 Rosetta 65

7.4 Squirell 66

7.5 Tacitus 67

7.6 Trains 68

7.7 Verbmobil 69

8 Evaluation of Two Systems 71

8.1 The Core Language Engine 71

8.1.1 Phenomena 72

8.1.2 Generalized Quantifiers and Scope 72

8.1.3 Computationally Relevant Aspects 74

8.2 Verbmobil 76

8.2.1 Phenomena 76

8.2.2 Computationally relevant aspects 79

8.2.3 Underspecification 79

9 Classification Dimensions 81

9.1 Introduction 81

9.1.1 The structure of the chapter 81

9.1.2 Semantic chronology 82

9.1.3 Interest Groups 85

9.2 Foundational Issues 86

9.2.1 Motivations 86

9.2.2 Philosophical issues 87

9.2.3 Mathematical issues 87

9.3 Practical Issues 88

9.3.1 Computational Issues 88

9.3.2 Empirical Issues 89

9.4 Classification Questionnaire 89

9.5 Unified views 115

9.5.1 Contextually Determined Reference Times 116

9.5.2 The Treatment of “Negative Events” 117

10 Annotated Text 119

10.1 Introduction 119

10.2 A Sample Text 119

10.3 A Syntactic Analysis 120

10.4 In Depth Description 121

10.5 Annotation 124

11 What can’t yet be done 135

11.1 Problems for the future 135

11.1.1 Contextual Reasoning 135

11.1.2 Spoken language input 136

11.1.3 Dialogue 137

Evaluating the State of the Art

The FRACAS Consortium

Robin Cooper, Dick Crouch, Jan van Eijck,
Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp,
Manfred Pinkal, Massimo Poesio, Steve Pulman

Abstract

In this document we discuss the themes of the interface to semantics, underspecification, contextual reasoning, inference and evaluation (in the sense of determining whether something is true or false), and lexical semantics. We then present a brief survey of some implemented systems (other than those described in D9) that are based at least in part on some of the approaches to semantics that we have discussed elsewhere. To prepare the ground for later comparisons we discuss the various dimensions along which semantic theories might be classified. We also give a brief selection of one or two areas of semantic description where, despite differences in theoretical stance, different semantic theories nevertheless show a remarkable degree of convergence in the way phenomena are handled. In order to ground discussion of the various themes and approaches in this document we include an annotated text ("Eurodisney") to illustrate the range and variety of semantic phenomena to be found even in the most simple and banal type of newspaper article. We classify the phenomena so as to give some idea of what is within the state of the art, and what areas still require a good deal of research. The concluding section amplifies this latter theme, trying to summarise the future directions that computational semantics might need to take in order to achieve enough progress to proceed towards the realisation of some of the goals sketched out earlier in this document.

Chapter 1

Preamble

Formal semantic treatments of NL phenomena look terrifyingly abstract, with lots of Greek letters, subscripts, fancy fonts and notations. This isn't typographical terrorism, but an attempt to be as mathematically precise as possible about the way to interpret natural language constructions. (Most of the time. Sometimes it is just camouflage for a bad or completely obvious idea.)

Putting things at this level of precision is analogous to the program specification step in software development: one wants to be absolutely sure of the detailed functional requirements for a particular program, preferably to a level of detail that would allow some formal verification and validation techniques to be used, so that you don't waste time coding things that don't do what they should. When you've got that level of specification sorted out, you can proceed to encode the resulting program in whatever the most convenient or efficient form is: this may not look much like the original specification. In order to know that the encoding preserves the intended interpretation of the original, one needs to have an unambiguous way of talking about the intended interpretation of the original: that is another reason why formal semantics spends such a lot of time defining the precise way in which various symbols get associated with elements of a model-theoretic universe.

Computational semantics can therefore be thought of as applied formal semantics. It adds to formal semantics a concern for computational properties and practical efficiency: that is to say, the extra requirement of choosing representations and algorithms that are not just mathematically coherent but are capable of supporting efficient processing. Computational semantics is also concerned with the practical issues of linking semantic interpretation with all the other aspects of language needed for a fully functioning implementation: deriving semantic interpretations from analysed sentences, relating them to contextual or dialogue properties, and eventually relating completed interpretations to the representations needed for computational applications.

Computational semantics is a relatively young sub-discipline, drawing on work in linguistic and programming language semantics, as well as computational techniques from AI and knowledge representation, formal methods, and theorem proving.

1.1 Practical applications of semantics

If computational semanticists had been able to solve most of the problems that interest them, and also been able to implement these solutions in a reasonably efficient and robust manner, what positive practical benefits would accrue?

It is common for people within NLP to express the belief that performance on tasks like message routing, textual information retrieval, or translation would improve if a reasonable depth of semantic processing could be carried out. The first two of these tasks (and to some extent the third) can actually be carried out quite well using statistical or pattern matching techniques that do not involve semantics in the sense we are assuming. It is in fact an open question for information retrieval and message routing as to whether semantic processing will improve on statistical methods: it is easy to devise artificial examples which require semantic processing for their correct treatment, but such examples may not occur at all in real life, or may occur so infrequently that failure to handle them correctly makes little difference to overall performance.

For **translation**, it is however reasonably clear that accurate semantic processing can only improve matters. This is especially so in the case of spoken language translation, for spoken language is inherently more context-dependent than most forms of written language. Producing the correct translation, even between closely related languages, often depends on being able to carry out reference and ellipsis resolution, interpretation of vague relations, and all the familiar panoply of semantic phenomena.

There are some tasks that cannot be carried out at all without semantic processing of some form. One important example application is that of **database query**, of the type chosen for the ATIS task ([DARPA, 1989]). For example, if a user asks "Does every flight from London to San Francisco stop over in

Reykyavik?" then the system needs to be able to deal with some simple semantic facts. Relational databases do not explicitly store propositions of the form 'every X has property P' and so a logical inference from the meaning of the sentence is required. In this case, 'every X has property P' is equivalent to 'there is no X that does not have property P' and a system that knows this will also therefore know that the answer to the question is 'no' if a non-stopping flight is found and 'yes' otherwise.

For restricted sublanguages of English, accurate semantic processing would make it feasible to **input data** in textual form rather than directly in some database formalism. For example, one could fill in an electronic "form" with a mixture of figures and short sentences or phrases, and have the results translated into some relational database format automatically. Such an application is in fact just about within the state of the art, for limited sublanguages of English and for simple domains of application (e.g. medical records, sales orders, customer complaints).

Under similar restrictions, it is feasible to think of **knowledge acquisition** or **system documentation** proceeding in a similar way. In addition to the end result of having the content represented in a formal way, enabling further computational processing of whatever type is required, the interactive process of disambiguation that semantic processing would make possible would be a valuable aid in eliminating unintended interpretations. For some applications, it is also possible that **knowledge representation** and reasoning carried out directly in a restricted natural language would be possible: see e.g. [McAllester and Givan, 1992]

Natural language generation is an area of practical application that is currently under-exploited. There are many domains where a natural language rendering of information represented in a non-linguistic form would be of practical value. Some examples are: **verbal summaries of tables** of numerical (e.g. financial) data; **system reports** of the internal state of some machine (e.g. in the case of breakdown); **paraphrases or justifications of decisions** taken by rule-based systems (e.g. expert or planning systems); **on-line documentation** or system manuals which link text to diagrams or pictures, and which may also be linked to e.g. spare part inventories or maintenance schedules (e.g. [McKeown *et al.*, 1994]).

Clearly for such a tool to be useful, the semantic content of the text generated should be an accurate expression of the non-linguistic information from which it is derived. To illustrate, if a database lists a 10am flight from London to Warsaw on the 1st-14th, and 16th-31st of November, then it is more helpful to answer the question "What days does that flight go?" by 'Every day except the 15th' instead of a list of 30 days of the month. But to do this the system needs to know that the semantic representations of the two propositions are equivalent.

1.2 Overview of this document

In the remainder of this document we discuss in a little more detail the themes of the interface to semantics, underspecification, contextual reasoning, inference and evaluation (in the sense of determining whether something is true or false), and lexical semantics. We then present a brief survey of some implemented systems (other than those described in D9) that are based at least in part on some of the approaches to semantics that we have discussed elsewhere.

To prepare the ground for later comparisons we discuss the various dimensions along which semantic theories might be classified. We also give a brief selection of one or two areas of semantic description where, despite differences in theoretical stance, different semantic theories nevertheless show a remarkable degree of convergence in the way the most important phenomena are handled.

In order to ground discussion of the various themes and approaches in this and earlier deliverables we include an annotated text ("Eurodisney") to illustrate the range and variety of semantic phenomena to be found even in the most simple and banal type of newspaper article. We classify the phenomena so as to give some idea of what is within the state of the art, and what areas still require a good deal of research.

The concluding section amplifies this latter theme, trying to summarise the future directions that computational semantics might need to take in order to achieve enough progress to proceed towards the realisation of some of the goals sketched out earlier in this introduction.

Chapter 2

The Interface to Syntax

2.1 Introductory remarks

The principle of compositionality expresses a property of the relation between the syntactic and semantic dimensions of expressions in (natural or artificial) languages. For obvious reasons (to do with e.g. methodological considerations and arguments related to issues such as learnability etc.) we would like this relation to be systematic and finitely statable. Compositionality in its various incarnations is a *guiding principle* in providing such a statement.[1] In its classical formulation the principle says that the meaning of a complex expression is determined by the meanings of its constituents and the way in which they are syntactically combined. This can be made precise in a number of ways. One version, sometimes referred to as "strict" or "rule-to-rule" compositionality, requires that each rule of syntactic composition be paired with a corresponding rule of semantic composition which computes the semantic value of the syntactic compound from the semantic values of its syntactic components. In addition, particular versions of this require that the composition process be defined in terms of function application(s) and further that for each local subtree there is one daughter whose semantic value provides the function applied to the semantic values of its other siblings. This is sometimes referred to as "function compositionality". Compositionality of the kind outlined above holds the considerable advantage that (sub)constituents in a derivation as well as the composition operation are fully interpreted. In actual work compositionality

[1] An alternative, finite and systematic statement of this relation can be provided e.g. in terms of the DRS-construction algorithm in the original formulation of DRT.

is often further constrained by an attempt to assign reasonably intuitive and plausible semantic interpretations to the subconstituents of a complex expression.

The compositionality principle or rather compositionality requirement can "trivialize" in the following sense: since compositionality expresses a property of the relation between the syntax and semantics of expressions it completely depends on the particulars of the given syntax and semantics pair. If we are free to vary the syntactic analysis and/or semantic values (i.e. denotations) associated with syntactic components it is always possible to come up with a compositional analysis for some fragment under consideration. In the discussion below we always assume that the term "compositionality" is used in a non-trivial sense.

Certain phenomena in natural language do not seem to lend themselves easily to a compositional treatment. There are several classes of non-local semantic phenomena which require reference to information encoded in arbitrarily distant parts of the syntactic representation, in order to achieve an appropriate interpretation for a given constituent. Quantifier scope, anaphora, and ellipsis are among the most important. Instead of a straightforward compositional treatment, representational approaches to the relation between syntactic and semantic structure have been proposed, which take into account the interaction of distant portions of semantic information, as well as non-local syntactic information constraining the interaction.

Several alternative solutions have been proposed in theoretical and computational semantics to model the basic mechanism of local meaning composition, and at the same time meet the requirements arising from non-local semantic phenomena. We will first discuss basic alternatives on the level of semantic formalisms and then look at the question of how the syntax-semantics interface can be realized in NL systems under a software-theoretic perspective.

2.2 Modular and Integrative Approaches

So-called *integrative approaches* provide a single uniform data structure for the representation (and manipulation) of different levels of representation. Different parts of this data structure can easily be accessed and interrelated. Levels of representation are usually constructed simultaneously and often in terms of general (constraint resolution) operations. *Modular approaches* offer specially tailored formalisms for each linguistic level of representation and are often (though not necessarily) wedded to sequential processing architectures where levels of representation are constructed in some particular order.

The problem with integrative approaches is that since in general the formalism used is “multi-purpose” it tends to be maximally general and powerful and rarely geared to the needs of some particular level of representation. Modular approaches, on the other hand, pose the additional task of providing an architecture that allows appropriate communication and cooperation between the levels, the data structures and operations used.

The classical contribution of theoretical semantics to the syntax-semantics-interface problem is Montague Grammar [Montague, 1970; Montague, 1973]. It is a clear example of a modular solution, providing separate data structures and operations for syntax and semantics. Montague’s syntax is a version of categorial grammar. The semantic representation language is Intensional Logic, a version of the typed λ-calculus. It is used for the representation of meanings, and at the same time as a “compositionality machine”: λ-abstraction and β-reduction steer linguistic substructures into place. The interaction between syntax and semantics is set up in a way that compositionality and interpretability of the composition process is guaranteed: semantic types and operations uniquely correspond to syntactic types and operations respectively; intermediate representations are denotationally interpretable.

There are, however, problems associated with Montague’s approach: Intensional Logic (IL), and, indeed, any standard logical syntax, has certain built-in restrictions; for example, terms have a fixed arity and take their arguments in a fixed order. Moreover, the mode of combination used by Montague — namely, typed λ-abstraction coupled with β-reduction — is inherently rigid; the order in which β-reductions must be performed is fixed ahead of time. While these restrictions are not particularly problematic in languages such as English where surface structure provides a good approximation to semantic representation, they pose difficulties when analysing languages where this is not the case.

In particular, problems arise with non-local semantic phenomena. Intersentential relations, such as discourse anaphora and ellipsis, are ignored in the original Montague fragment. Sentence-internal non-local phenomena are treated in the syntax part of Montague grammar: The input to semantic interpretation is a disambiguated structure (an abstract syntactic level similar to LF in the GB framework, cf. [May, 1985]), with quantificational NPs located in their intended scope position, and pronouns co-indexed with their binding NPs. Convincing as the compositional core of Montague’s solution is, it is widely agreed that this treatment of non-local phenomena is theoretically unsatisfactory and practically inapplicable.

Computational linguistics has favoured integrative solutions to the problem of level interaction. The idea goes back to LFG [Kaplan and Bresnan, 1982; Halvorsen, 1983], and has been taken up by other frameworks, notably HPSG [Pollard and Sag, 1994]. A single formalism is used to describe linguistic structures, operations and constraints on all levels. The formalisms used are generally referred to as "unification-based". In most cases they provide some kind of feature specification language and a logic for the language. Feature-structure representations are less rigid than type-theoretical ones, and unification is an intuitively appealing universal combination method. Also, integrative frameworks encourage combination of operations concerning different linguistic levels and formulation of multi-level constraints. However, by choosing a general and flexible formalism instead of one which is genuinely semantic, transparence and precision on the side of semantic representations may be lost.

Feature structure representations (or the corresponding formulas in the feature-logic) of semantic representations may not be directly interpretable in the sense that unlike type-theoretical expressions in Montague's IL they do not wear their intended *semantic* interpretation on their sleeves. Of course if the feature structure formalism itself is interpreted the representation is interpreted in terms of the formalism. This interpretation, however, will in general not correspond to any intuitive interpretation of the semantic contribution of some constituent in a Montagovian set-up[2] but rather be dictated by the particulars of the feature structure representation and the combination process (i.e. "unification").[3]

In the following, we will concentrate on the discussion of modular solutions for a syntax-semantics interface. It should be pointed out, however, that the difference between "modular" and "integrative" is not that clear-cut on the implementation level. At least the target structures in integrative feature-based systems are usually translatable into logical, i.e. denotationally interpretable

[2] Often such formalisms are partially interpreted in the sense that that a straightforward "semantic" interpretation for some feature structure rendition of a semantic representation is available at the completed sentential level.

[3] This problem is visible in e.g. what has been referred to as a mixing of "object"- and "meta"-level uses of variables in feature structure representations of semantic representations. On the one hand variables are interpreted as variables in the underlying feature logic (with its particular scope and binding definitions etc.) and as variables in some "intended" sense in what is regarded as a semantic representation. This situation is already present in e.g. Prolog implementations. If used properly - ie. if one is aware of the limits of this approach (c.f. see the remarks on the simulation of β-reduction below) - this dual function of variables does not strike us as particularly troublesome.

formulae. On the other hand, most existing modular systems are implemented in unification-based languages (cf. [Alshawi, 1992] for QLF, [Bos *et al.*, 1994a] for Verbmobil), and the binding mechanism is mimicked by unification. Therefore they are subject to the same kinds of inadequacy, which have been pointed out for integrative feature-based approaches. Problems may occur if λ-terms and particularly λ-variables are modelled (rather: approximated) in terms of the terms and the variables of the feature description language and β-reduction is implemented in terms of first-order unification. In such a context wrong results can emerge if β-reduction is carried out with a higher-order variable which occurs more than once in the body of the λ-expression [Moore, 1989]. There is a methodological difference, however, to "naive" unification-based systems: The choice of possible semantic representations and operations is constrained by the logic of the modelled semantic formalism, and one-to-one translation between the two modes of representation is possible, as long as the limitations of the simulation are respected.

2.3 Modelling Non-local Semantic Phenomena

The formalism of Montague Grammar appears to establish too rigid a connection between syntactic and semantic structure. Retracting to a weaker universally applicable formalism seems to be no serious solution, however. In the following, we will discuss several methods to adapt the modular approach to the requirements of natural-language semantics, especially to the needs of non-local semantic phenomena. We will first inspect solutions that have been proposed for two special kinds of non-local phenomena, quantifier scope and anaphora, and then discuss the possibility of a more general strategy.

For quantifier scoping, the so-called "*Cooper Storage technique*" has become a standard tool in many implementations [Cooper, 1983; Keller, 1988]. Essentially, it is a way of representing and manipulating quantifiers that allows scope choices to be deferred. Positions which require quantifiers are noted (a place-marking variable is introduced, and abstracted across), but the quantifier is put "on ice" until the scopes can be resolved (typically, at a category of type "proposition"). The semantic value of a quantified sentence can be computed (locally and bottom-up) on the surface constituent structure rather than on some abstract disambiguated or LF syntactic representation. The price to be paid is that the semantic information attached to a syntactic constituent is not a simple straightforwardly interpretable logical expression anymore, but a structured object consisting of an open expression (containing free variables), and a set of quantifiers to be applied.

The problem of a proper description of *anaphoric reference* led to a drastic deviation from Montague's concept of compositionality: Discourse Representation Theory in its standard version [Kamp, 1981b; Kamp and Reyle, 1993] is non-compositional in the sense that it employs a top-down construction algorithm for semantic analysis where most of the construction rules are not directly semantically interpretatable. [Johnson and Klein, 1986] present a declarative reformulation of "core"-DRT formulated in terms of constraints and constraint resolution which combines "bottom-up" and "top-down" information flow in terms of the threading method to pass non-local information (the set of currently available anaphoric referents - in fact the entire semantic representation) through the constituent structure of the sentence, to make it available at those nodes where e.g. anaphoric NPs are to be interpreted. The semantic contribution of constituents in a syntactic representation is defined in terms of "updates" of and "tests" on the threaded representations. In a way, their system can be seen as a computational (and representational) predecessor to Dynamic Predicate Logic: DPL models the transport of non-local anaphoric information by passing modified variable assignments between constituents [Groenendijk and Stokhof, 1991]. Lambda-DRT [Bos *et al.*, 1994b] combines the DRT representation format with typed λ-calculus, thus providing the DRT formalism with the compositional mechanism of Montague Grammar. [Muskens, 1994] describes a similar solution for a "compositional discourse representation theory".

Some formalisms for underspecified semantic representation (cf. [Alshawi, 1992]) in a certain sense confine the non-compositionality problem to particular stages of a derivation. They divide the task of semantic analysis into two major steps, which may be carried out at different times: First, an underspecified representation is built using all kinds of locally available linguistic information. Second, this representation may be resolved and transformed into a fully specified standard representation.[4] This implies a corresponding division of labour in the task of treating non-local phenomena: First, underspecified representations for non-local phenomena (unscoped quantifier terms; unbound pronouns; unresolved ellipsis) are built up in a compositional way. The assignment of quantifiers to their proper scope positions, and anaphora resolution, is postponed and confined to the second phase of processing, where the underspecified representation is transformed to a logical formula.[5] Cooper storage can be

[4] UDRS construction as detailed in [Reyle, 1993a; Frank and Reyle, 1992; Frank and Reyle, 1994] defines an integrated framework and does not really presuppose a two-stage derivation of the kind outlined above. However, the framework is amenable to a two-stage architecture if for whatever reasons one should opt for such a set-up.

[5] The treatment of anaphora in implemented systems often follows this two-step policy. (A

viewed as a variant of such a technique.

A more general view on compositionality problems is opened by the concept of "glue languages" [Dalrymple *et al.*, 1993]. It is a generalisation on a number of dimensions of Montague's use of the λ-calculus. Glue languages perform much the same function as the λ-calculus did for Montague, but do so in a far more flexible way. The fundamental idea is very simple. Two types of language should be distinguished: a *semantic representation language* ([Dalrymple *et al.*, 1993] call this the *language of meanings*), and a *language for assembling meanings*, or glue language. The task of the semantic representation language is to represent the meaning of the sentence; the task of the glue language is to manipulate this representation. This is clearly related to Montague's use of the typed λ- calculus to manipulate IL expressions, but there are important differences. Firstly, the representation language and glue languages are distinct; in principle, the linguist could choose any two formalisms. This means that the linguist can select two formalisms each of which has properties optimal for the task at hand. Secondly, by separating the problems of *constructing* representation from the representation problem proper, they open the way to flexible, yet principled, treatments of scoping. In particular, the way is opened to treatments of scope via the notion of logical proof.

For their representation language, [Dalrymple *et al.*, 1993] choose a standard logical language (in fact, the language of first order terms). Thus the final representation will be an expression with well understood syntactic and semantic properties. For their glue language they choose the tensor fragment of first order linear logic (see [Girard, 1987]). Linear logic is a "resource conscious" logic: it is sensitive to the amount of information it consumes in the course of a proof. By representing the information contained in a linguistic structure as constraints in linear logic, it is guaranteed that *precisely* this amount of information is used to construct the semantic representation.

Glue languages allow to manipulate representations in a flexible, but principled way, for example by logical deduction in a suitable glue calculus. Manipulation does not have have to involve ad-hoc procedural stipulation. This makes the glue language approach a very interesting alternative to model the syntax-semantic interface, although the treatment of interesting cases like quantifier scoping is somewhat programmatic as yet.

theoretical account of a two-step treatment of anaphoric expressions can be found in [Sandt, 1992]).

2.4 Formalisms and Processing Models

In 2.2, we discussed the choice between integrative and modular approaches to syntactic-semantic analysis. The concepts, as we have used them so far, refer to the question of how many formalisms are used for modelling the different linguistic levels of syntax and semantics. By themselves, they do not fix the way the different modules interact to perform the task of processing natural language utterances. However, they are usually associated with different processing strategies: Integrative approaches support interleaved syntactic and semantic processing, with an unlimited flow of information between the different linguistic levels (cf. the "constraint pool" concept in [Fenstad *et al.*, 1987a]). On the other hand, the kind of modular approach as it has been put forward by GB oriented linguistics suggests a strictly sequential processing mode.

The association between modularity and sequentiality has been made a major topic by advocates of integrative solutions. However, integrative approaches are not only in danger to fall short of theoretical rigour, as has been pointed out in 2.2, but also tend to be difficult to handle by system developers, since they lead to large systems with complex internal dependencies. On the other hand, the connection between modularity and sequentiality is by far not as close as has been occasionally claimed. In the following, we try to give a slightly more detailed account of the connection between modularity and possible processing strategies.

Computational linguistics have often referred to Chomsky's Government-and-Binding framework as a deterrent example of an unrealistic processing model: In fact, the information flow between the modules is highly restricted in GB. Semantic interpretation rules operate on the output of syntactic analysis in a strictly local and deterministic way: Semantics has access only to that kind of syntactic information which becomes locally available when the syntax tree is trespassed by the semantic interpretation process. Syntax, on the other hand, never has access to information of the semantic level: the flow of information is unidirectional. It is important to notice that what we have said so far concerns only the logical dependencies between the modules. It does not directly constrain the temporal order of processing. For illustration, consider Montague Grammar: All of the properties just mentioned in connection with the GB framework apply to Montague's rule-to-rule account of syntax-semantics interaction as well. Montague's approach, however, suggests a completely different way to synchronize syntactic and semantic analysis: each application of a syntactic rule immediately triggers the execution of the associated semantic rule. In fact, it is the independence of specific processing modes which makes compositional semantic approaches so attractive for NL applications.

Strictly speaking what we have said in the last paragraph holds for that part of semantics which is given a straightforward compositional analysis. Non-local phenomena in natural language semantics sometimes complicate the relation between formalism and processing strategy. The solution chosen by Chomsky as well as by Montague is the introduction of an additional syntactic level of "Logical Form" or "Disambiguated Structure", which mediates between certain non-local phenomena and the locality requirement of simple compositional approaches. Often it is this introduction of an intermediate level which makes it difficult to avoid a sequential architecture and leads to problems with respect to (ambitious goals such as) cognitive adequacy and (sometimes perhaps - although this can be disputed) computational efficiency.

Some of the more flexible and "less compositional" approaches to semantic analysis described in 2.3 work without such an additional level. However, they are faced with the problem that non-local semantic operations are often constrained by syntactic information which likewise is not locally available. Scope islands for quantifiers are the most familiar type of such non-local constraints. More complex configurational constraints governing the relative scope order of quantifiers are investigated in [Frey, 1993]. In the field of anaphoric binding, the Weak Cross-Over constraint has been discussed at length [Pinkal, 1991a]. A striking example for the insufficiency of simple conventional models of syntax-semantics interaction is ellipsis. It seems that neither a purely syntactic (LF) treatment [Fiengo and May, 1991] nor a semantic account which disregards syntactic information are able to tell the true story about ellipsis interpretation: An interleaved processing mode is required which allows linguistic information of different kinds and from different places to contribute simultaneously to the analysis of natural language utterances [Gardent, 1991].

One option to solve these problems is to enrich intermediate semantic representations (e.g. stored quantifiers, pronominal variables to be bound) with the kind of syntactic information which may become relevant later on (this is, e.g., basically the CLE solution [Alshawi, 1990]). In the following section, we will describe proposals to relate system components in a way that allows controlled and systematic exchange of information between the components.

2.5 Modular Systems for Syntactic-Semantic Processing

Central to the software-theoretic notion of modularity is the idea that representations and processes particular to some component of a system should not matter to other components. In particular one module may be substituted for another with identical outside appearance and functionality without affecting the rest of the system. Central to this idea is the concept of abstract data types. An abstract type (ADT) provides an internal organization of data together with an external interface to access them. The details of the internal representation are hidden from the outside world - no information internal to a module may be inspected or manipulated directly.

As we pointed out in section 2.4 above, computational models of semantic interpretation must make use of information from other levels of linguistic structure. In the following, we consider computational approaches which make this information available by treating the relevant modules as ADTs.

In [Johnson and Kay, 1990] an approach is proposed, in which one can build up semantic representations according to different semantic formalisms, e.g. predicate logic (PL), situation semantics, or DRT. Johnson/Kay define a set of abstract "semantic constructors", which get different definitions for each of the semantic theories. They illustrate their approach in the framework of DCGs. Here is an example.

```
n1(X^S) --> n(X^S1), rc(X^S2), {conjoin(S1, S2, S)}.
```

The `conjoin/3` constructor would be defined simply as conjunction, if one were using PL. In DRT, it would be defined as taking the union of the sets of markers and of conditions, respectively.[6] Effectively, Johnson/Kay treat *semantics* as an ADT from the perspective of syntax. Their strategy is a very interesting one. However, they fail in their attempt to insulate the grammar writer from the details of the possible semantic target theories, about which s/he must still know a great deal. Consider the following, more complex example, showing the lexical entry for every:

```
determiner(every, Res0^Scope^S) :-
```

[6] Johnson/Kay also provide a situation semantics definition.

```
compose(S1, S2. S),
subordinate(Res, ResName, S1),
compose(Res0, Res1, Res),
subordinate(Scope, ScopeName, Res1),
atom(ResName ==> ScopeName, S2).
```

The two calls to the `subordinate/3` constructor are needed to introduce some hierarchical structure into the representation, which is used by DRT (but not by PL) for the purposes of accessibility constraints. Note also that one has to introduce names for the restriction and scope of the quantifier (`ResName`,`ScopeName`) because situation semantics distinguishes a propositional content from the situation in which it is true (although PL does not). As far as a productive division of labour is concerned, it seems wrong to require the grammar writer to know several semantic theories in detail. Concerning Johnson/Kay's particular choice of constructors, they are primarily oriented towards neutralizing differences between the way semantic target representations are written, but cannot really be said to embody *semantically* primitive notions.

Another strategy is used in the Verbmobil System [Bos *et al.*, 1994a]. In deriving semantic representations, certain facts about the syntactic analysis must be known. By treating the *syntax* as an ADT, the particular form in which this information is encoded in different grammars can be disregarded. In Verbmobil, this approach is realized in terms of *macros* in a feature-based language, which provide atomic information about linguistic entities or information about relations between such entities.

case(Sign,Case)
head(Sign, Head)

Note that the Verbmobil grammar is a version of HPSG. Semantic formulae occur as the value of a SEM feature, and are represented in the same underlying feature language as the syntax. The semantics uses a compositional, DRS-based formalism, which is likewise implemented in the feature language. Thus, the system is integrative on the level of the implementation language. This is no hindrance to a modular approach, as the example of Verbmobil shows. On the contrary, integration makes it easy to have two-way (or multi-way) communication between linguistic levels by just defining the appropriate predicates in the underlying language. Modularity basically requires introducing an appropriate level of abstraction from the underlying representation.

In another system, namely the SCOLD system developed in the SAMOS project, the ADT-approach has been implemented in a non-integrated environment [Millies and Pinkal, 1993]. In SCOLD, a single semantic interpretation component has been interfaced with different syntactic formalisms, in particular GB and LFG. The interface is realized by predicates and functions, which abstract from the differences between the syntactic representations. The approach has been carried further than in Verbmobil insofar as the relevant domain of interpretation is not fixed but depends on syntax (parts of f-structures in LFG, resp. sets of tree nodes in GB). Also, it is possible to use non-lexical information in the determination of functor-argument relationships.

The semantic interpretation component in the SCOLD system is itself divided into two parts, taking up the ideas of Johnson/Kay: There is an interpretation algorithm which accesses syntactic information via the interface predicates, and calls semantics operators to build semantic representations. In contrast to Johnson/Kay, these operators are genuinely semantic, specifying operations like abstraction or generalized functional application, which is the main operation used to combine semantic objects. Although in fact these operators build representations of λ-DRT, their definitions could be changed to build other kinds of representations without changing either the syntax or the interpretation component. SCOLD assumes only that semantic representations are build in a compositional fashion using techniques from the λ-calculus.

There are certain limits to the factual independence of modules. In some cases, a grammar formalism may not directly provide the syntactic information necessary to implement some constraint on interpretation. Therefore, a practical advantage can be gained from working with one syntax rather than another. Sometimes the problem is deeper, i.e. with re-entrancy in LFG. In order to derive the correct representations for f-structures involving e.g. control constructions, it is necessary to distinguish the controller from the controlled element. But f-structure simply does not contain this information, it is hidden in the syntax-rule annotations.

It is probably unrealistic to expect the modular approach to support unconditional exchangeability and portability of modules across system and theory boundaries. However, modularity in the software-theoretic sense makes components of NL systems relatively stable and independent of changes in other components, and thus, e.g., supports the distributed development of large systems.

Chapter 3

Underspecification

3.1 The Case for Underspecification

The (literal) meaning of an utterance depends both on its form and on its context of utterance. In the absence of context, sentences are frequently ambiguous or vague, and this situation can sometimes persist even when context is taken into account.

'Generate and test' has been a traditional approach to the problem of vagueness and ambiguity. Syntax and semantic interpretation rules based on syntactic structure are set up in such a way as to generate all possible readings. These readings are tested against some (typically ill-defined) notion of contextual plausibility until all but the intended interpretation are eliminated.

Generate and test is widely held to be computationally impractical and, perhaps, theoretically indefensible. Within many current formal theories of syntax and semantics, intuitively straightforward sentences can give rise to thousands, if not hundreds of thousands of ambiguities.[1] In the case of vagueness, it is arguable that there may be an infinite number of more precise ways of spelling out possible interpretations.

To avoid problems with combinatorial explosion, many have proposed the use of underspecified semantic representations.[2] Built up on the basis of the syntactic structure of a sentence, these representations leave contextual components of meaning unspecified. Contextual resolution fills in these areas of underspecification; as opposed to eliminating fully specified but contextually implausible interpretations as in generate and test.

Underspecified representations were originally conceived as a way of solving a problem in system implementation (see, e.g., [Woods, 1977]), namely, separating 'context-independent' from 'context dependent' aspects of the interpretation, thus making either part reusable for different applications. As such, underspecified representations were an intermediate stage on the way towards obtaining a fully specified representation of meaning. Recently, the idea has been gaining ground that underspecified semantic representations are valuable in their own right: that it might be possible to perform, e.g. inference, directly on underspecified meanings. Moreover, if one assumes that humans represent meanings in some underspecified form, it would explain why many technically ambiguous sentences are not *perceived* as ambiguous.

In the following sections we will mention a number of issues relating to underspecification as a prelude to drawing up a set of dimensions for evaluating different treatments of underspecification.[3]

[1] For example, (3.1) may well be assigned hundreds of distinct parses, and (3.2) perhaps hundreds of thousands of distinct scopings:

(3.1) We should move the engine at Avon, engine E1, to Dansville to pick up the boxcar there, then move it from Dansville to Corning, load some oranges, and then move it on to Bath.

(3.2) A politician can fool most voters on most issues most of the time, but no politician can fool all voters on every single issue all of the time.

[2] E.g. the 'Logical Form' proposed by Schubert and Pelletier [1982], the two level semantics of the PHLIQ1A system, the 'Situation Schemata' of Fenstad *et al.* [1987b], and the 'Logical Form' discussed in Allen's textbook [1987]; more recently, the constraint-based semantics of Nerbonne [Nerbonne, 1991], Quasi Logical Form [Alshawi and van Eijck, ; Alshawi and Crouch, 1992; Alshawi, 1992], Reyle's Underspecified DRT [Reyle, 1993b] and Poesio's 'Conversation Representation Theory' [Poesio, 1994].

[3] Part of the solution to the combinatorial explosion problem may be that some forms of ambiguity at least are solved *locally* and *incrementally*, and therefore the dreaded explosion of readings never occurs, since at any moment, only a score or so of readings need to be

3.2 Background

3.2.1 Ambiguity

Underspecified semantic representations have been proposed as a solution to the problems for both psychological theory and NLP implementors generated by the proliferation of ambiguities and vaguenesses in the meanings of sentences.

This proliferation can have a variety of sources: structural/syntactic ambiguity; lexical ambiguity; different relative scopes of operators and quantifiers; different antecedents for anaphors and ellipses; implicit relations, e.g. possessives, compound nouns; vague relations and properties. It is an open question whether a single approach to underspecification can be applied to vaguenesses and ambiguities arising from all these sources. For example, syntactic and semantic ambiguity are often treated separately; a sentence that is N-ways syntactically ambiguous will be assigned N distinct syntactic analyses, and from these N (probably distinct) underspecified semantic representations are built.

Several definitions of, and distinctions between, ambiguity and vagueness have been proposed in the literature [Pinkal, 1991b]. For semantic ambiguity, it is useful to distinguish between two types of approach: structural and semantic.

A structural view of ambiguity states that: *An expression is ambiguous iff the expression can accommodate more than one structural analysis.* ([Gillon, 1990], p. 400). In a theory such as Government and Binding [Chomsky, 1981; Haegeman, 1991], each interpretation of a sentence is a quadruple ⟨PF,DS,SS,LF⟩, where PF characterizes the phonetic interpretation, DS the predicate/argument composition of the sentence, SS its structural analysis, and LF its logical analysis. Under this characterization, a sentence is ambiguous iff it can be charac-

considered. This hypothesis is backed up by psychological evidence in the case of lexical ambiguity [Seidenberg *et al.*, 1982; Swinney, 1979]. Structural and scopal ambiguities cannot be immediately resolved, however: in a sentence such as *John gave a present to each child*, for example, the indefinite *a present* takes narrow scope with respect to the quantifier *each child*. The interpretation of the sentence must therefore either remain partially underspecified until the quantifier is processed, or be revised when the quantifier is encountered. Similarly, when processing a sentence such as *John always invites MARY to the movies*, whose preferred interpretation is that whenever John goes to the movies, he invites Mary, the restriction of the adverb of quantification *always* is not encountered until the PP *to the movies* is encountered. Also in this case, it appears that even assuming an incremental processing strategy, the interpretation cannot be completely specified until well after the quantifier is encountered. This suggests that an incremental account of discourse interpretation, as well, must be supplemented with a theory of underspecification.

terized by two distinct quadruples.

This kind of characterization works fairly well for syntactic and phonetic ambiguities, but is problematic for semantic ambiguity. One problem is that a purely syntactic analysis of meaning introduces spurious distinctions: thus, for example, it is necessary to say something about invariance under renaming of variables to avoid predicting that a sentence such as *Every man left* is infinitely ambiguous because all LFs of the form $[_{\mathrm{S}}\ [_{\mathrm{NP}}$ every x man$][_{\mathrm{x\ left}}\ ]]$ for any choice of the variable are appropriate translations of the sentence. For a structural account of semantic ambiguity to work, there must be some kind of one-to-one correspondence between structures and meanings.

A semantic view of ambiguity frames ambiguity in terms of having distinct interpretations in some model according to a grammar. A string σ (a word or a larger constituent, such a sentence) of language $\mathcal{L}$ is ambiguous with respect to a translation function τ that maps expressions of $\mathcal{L}$ into objects of a model $\mathcal{M}$, if $\tau(\sigma) = \{\tau_1, \ldots, \tau_n\}$, where $\tau_1, \ldots \tau_n$ are distinct objects of $\mathcal{M}$.[4]

For example, the word *croaked* maps onto two distinct denotations: the set of objects that made frog-like sounds, and the set of objects that died. Let us call the semantic correlates of sentences (utterances) propositions. When a lexically ambiguous item like *croaked* is included in the sentence *Kermit croaked*, the sentence is ambiguous between a proposition saying that Kermit made a frog-like sound and a proposition saying that Kermit died.

This definition of semantic ambiguity relies on some assumptions both about the kind of object that is used to model the notion of proposition and about the kind of object that the grammar assigns as a meaning to a sentence constituent. If we identify propositions with truth values, and we take the meaning of a sentence to be a proposition, the sentence *Kermit croaked* turns out to be unambiguous with respect to a model and a translation function according to the definition above if Kermit has both the property of dying and the property of producing a frog-like sound in that model, or if he (it) has neither property. The definition above relies on the adoption of a more fine-grained notion of proposition is adopted, such as the one adopted in the theories derived from Montague Grammar, where

[4] In practice, the translation function τ is typically formulated as a mapping from expressions of $\mathcal{L}$ to expressions of a 'translation language' $\mathcal{TL}$, such that each expression of $\mathcal{TL}$ denotes a single object in $\mathcal{M}$. Typically, $\mathcal{TL}$ is a logical language with the usual ingredients such as predicates, connectives, etc. This fact shouldn't however lead to the conclusion that the 'semantic' definition of semantic ambiguity is essentially identical to the 'syntactic' definition: the former, unlike the latter, does not depend on the syntax of $\mathcal{TL}$, but only on its interpretation.

propositions are functions from possible worlds or situations to truth values;[5] as well as on the assumption, common in modern semantic theory, that the meaning of a sentence (the value assigned to it by the translation function τ) is a function from contexts (or discourse situations) to propositions.[6]

3.2.2 Vagueness, Indefiniteness and Uninformativeness

A semantic definition of ambiguity makes it hard to distinguish from vagueness. Both vague and ambiguous sentences are ones that can have more than one interpretation, so that as Pinkal [1985] puts it

> in certain situations, despite sufficient knowledge of the relevant facts, neither "true" nor "false" can clearly be assigned as its truth value.

Pinkal goes on to argue that vagueness and ambiguity are both cases of indefiniteness. Indefinite sentences can be made more precise, and in doing so can be made either true or false. A 'precisification' ordering over interpretations can be defined, such that p is more precise than q iff p is true (false) under all states of the world (models) in which q is true (false), and p is true or false in some models in which q is indefinite.

The difference between vagueness and ambiguity, according to Pinkal, lies in whether the precisification ordering appears continuous or discrete. For a vague predicate like *green* or *fast* there is a continuum of possible precisifications, depending on the shades of green or degrees of speed countenanced. An ambiguous predicate, like *croaked*, has a much sharper division between its specific senses.

It is important to distinguish Pinkal's vagueness / indefiniteness from uninformativeness. A sentence is uninformative if while definitely true or false, it could be made more specific. Thus, *Kermit found a glove* is uninformative about whether it was a left-handed or right-handed glove. The predicate *glove* is not vague as to left- or right-handedness, but non-specific.[7] It hardly needs to be pointed out that just about every sentence is uninformative / non-specific in some respects.

[5] More complex notions of propositions, such as those used in Situation Semantics [Barwise and Cooper, 1993] or Property Theory [Turner, 1992] ensure an even finer grained distinction.

[6] For a discussion of these assumptions, see [Barwise and Perry, 1983] or [Kaplan, 1977], as well as chapter 2 of [Pinkal, 1985].

[7] Nerbonne [1991] by contrast does claim that *glove* is vague, though not ambiguous.

3.3 Approaches to Underspecification

3.3.1 Comparing Theories of Underspecification

There are two main dimensions along which an approach to underspecification can be classified: one of them is the theory of ambiguity underlying that approach, the other is the theory of discourse interpretation and disambiguation that is assumed.

The differences between the existing approaches to underspecification can thus be expressed in terms of the way they answer the following questions:

- What is the appropriate characterization of the notion of ambiguity? Is it distinct from the notion of vagueness?
- Are (some forms of) ambiguity left unresolved by humans? Should systems do the same?
- Are all kinds of ambiguity resolved in the same way? Is the disambiguation monotonic or non-monotonic?
- What kinds of information are encoded in underspecified representations?
- Does reasoning take place before a complete disambiguation has occurred?

3.3.2 QLF

Underspecification in QLF has already been covered to some degree in deliverable D8, so here we will merely address the questions above.

QLF does not draw a sharp distinction between ambiguity and vagueness. Vagueness is typically resolved through the choice of some contextually salient comparison class for predicates like *green* or *fast*, and in this respect does not differ greatly from the choice of contextual restrictions on quantifiers for, e.g. anaphora. QLF permits underspecification both in the meanings of various elements (e.g. pronouns, vague predicates), and the way in which meanings are combined (e.g. scope of quantifiers). As such it lends itself to both semantic and structural approaches to ambiguity.

While unresolved QLFs have a semantics, they do not have a proof theory. Therefore, if inference is to be done on the results of semantic interpretation, it pays to resolve ambiguities completely. The same goes for vagueness in QLF, though here it is extremely doubtful if humans ever do produce or require complete precisifications of vagueness.

In order to carry out disambiguation, inference on components of the QLF is required. However, it is usually possible to carry this out in such a way (bottom up resolution) that reasoning is only ever carried out using completely resolved subformulas of the QLF in question.

An important facet of resolution in QLF is its monotonicity: the order in which ambiguities are resolved does not affect the range of possible disambiguations. This is reflected in the uniform use made of unification as a tool for marking resolutions in QLF; while the techniques for selecting appropriate resolutions may differ from one kind of phenomenon to another, at the level of making QLFs more precise they are all dealt with in the same way.

Deliverable D8 argued that underspecified QLFs encode information about permissible semantic derivations, though this claim is perhaps debatable. As a result, QLFs inevitably include a good deal of syntactic/grammatical information, relevant to the choice of semantic composition.

Nerbonne's [1991] constraint-based semantics shares many properties of the QLF treatment of underspecification. However, by concentrating on the syntax of logical forms, it veers towards an overly structural account of semantic ambiguity. And one might also quibble over Nerbonne's treatment of vagueness, which in Pinkal's terms sounds more like uninformativeness.

3.3.3 Underspecified DRSs

Reyle's goal in [Reyle, 1993b] is to revise the DRT construction algorithm to make it functional—i.e., such that a single underspecified translation is assigned to an ambiguous sentence—preserving at the same time the capability of running inferences directly on these underspecified representations without considering disambiguated cases. Reyle is concerned in the paper with scopal ambiguity; the proposal has been recently extended to deal with the collective/distributive ambiguity.

Reyle notes first of all that DRSs could easily be represented in an alternative 'label-and-condition' representation. Thus, for example, the DRS in (3.4) could be equivalently represented as in (3.5), by assigning to each 'box' a *label* and by labeling each condition and discourse marker with the label of the box in which it occurs. E.g., the outer box could be given the label l1; the occurrence of a complex condition representing the quantifier *every* in the box l1 could be represented by a statement of the form l1:l2 ⇒ l3, where l2 is the label assigned to the restriction of the tripartite condition, whereas l3 is the label of the nuclear scope. The occurrence of the discourse marker x in l2 could be represented by a statement of the form l2:x, whereas the occurrence of the condition $\textsc{climbed}(x{,}y)$ in the nuclear scope can be represented by the statement l3:$\textsc{climbed}(x{,}y)$.

(3.3) Every kid climbed a tree.

(3.4)
$$\boxed{\quad\begin{array}{|c|}\hline \text{x} \\ \hline \textsc{kid}(\text{x}) \\ \hline\end{array} \quad\Rightarrow\quad \begin{array}{|c|}\hline \text{y} \\ \hline \textsc{tree}(\text{y}) \\ \textsc{climbed}(\text{x,y}) \\ \hline\end{array}\quad}$$

(3.5) l1:l2 ⇒ l3
l2:x
l2:$\textsc{kid}(x)$
l3:y
l3:$\textsc{tree}(y)$
l3:$\textsc{climbed}(x{,}y)$

Secondly, Reyle observes that the labelled representation can be generalized so as to leave the scope of various expressions unspecified, by adding the possibility of leaving the relation between the boxes only partially specified. For example, the interpretation of the sentence above in which the relative scope of the quantifiers isn't determined could be represented as in (3.6). The contribution of the universal quantifier *every kid* to the meaning of the sentence is represented by the first four conditions in (3.6), which assert that a complex condition is included in some box l0, that the restriction of the complex condition includes a discourse marker x and a property of that discourse marker, and that l0 must be equal to or be in the scope of the top box l1. The subsequent two statements represent the contribution of the indefinite *a tree* and assert that a box labeled l4 must include a discourse marker y and a condition restricting that discourse marker to take values ranging over trees. Finally, a box l5 is introduced that represents the contribution of the VP. Both the nuclear scope of *every kid* and the l4 box must be the same as, or take scope over, the box l5. In other words, every box introduced by an operator has both an upper and a lower bound.

(3.6) l0:l2 $\Rightarrow$ l3
l2:x
l2:KID(x)
l1 $\preceq$ l0
l4:y
l4:TREE(y)
l1 $\preceq$ l4
l5:CLIMBED(x,y)
l3 $\preceq$ l5
l4 $\preceq$ l5

Reyle provides an example showing how such representations could be produced by lexical specifications; work on using UDRSs as the semantic interpretation produced by an HPSG grammar is described in [Frank and Reyle, 1992].

Perhaps the most interesting aspect of the UDRS proposal is that Reyle proposes a complete and sound inference system for the first-order fragment of UDRSs. This system is a generalization of the system proposed in [Kamp and Reyle, 1991], and is based on the following definition of truth at a situation:

> ... We will assume that an ambiguous expression is true in a given situation in case one of its disambiguations is.

UDRSs define *partial* orders between scope inducing elements (such as quantifiers and logical operators) in a representation. Currently, partial orders between quantifiers can only be represented in the QLF format to a limited extent.[8] In contrast to QLF, in UDRT no provision is made for a treatment of vague relations (e.g. possessives, prepositions etc.). Both the QLF and UDRT approach are monotonic in the sense that they do not involve destructive operations on the representations, in the sense that the order in which representations are manipulated is not germane with respect to the resulting representation and in the sense that addition of further information narrows down available interpretations.

The UDRS representation is currently being used in several projects.

[8] E.g. a QLF formula which does *not* contain another QLF formula as one of its constituents cannot express that of three quantifiers Q_1, Q_2 and Q_3, Q_1 and Q_2 have scope over Q_3 while the scope relation between Q_1 and Q_2 is left unspecified.

3.3.4 Conversation Representation Theory

Poesio proposes in [Poesio, 1994] an account of disambiguation and discourse interpretation based on a distinction between *semantic ambiguity* and *perceived ambiguity.* The theory of ambiguity proposed by Poesio consists of two main parts:

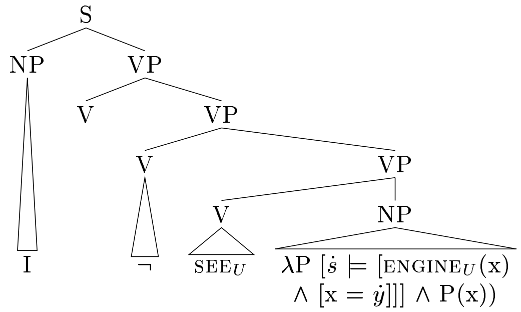
1. an *underspecified* language which can be used to code semantic ambiguity implicitly, thus eliminating the need to generate all semantic interpretations; and
2. a theory of the disambiguation process that may result in a perceived ambiguity.

Poesio is concerned with scopal and referential ambiguity. A generalization of Montague Grammar similar to the one in [Cooper, 1975] is proposed, whereby lexical items and other sentence constituents are assigned a set of interpretations by a grammar.[9] A sentence constituent is semantically ambiguous under a grammar G if G assigns a non-singleton set of interpretations to that sentence constituent. Where Poesio's approach differs from Cooper's is that an 'underspecified language' is introduced that can be used to assign a single translation to ambiguous expressions of the language. For example, assume that the word *croak* is ambiguous between two type $\langle e,t \rangle$ translations: the property CROAK$_1$, which is true of objects that produce a sound like that produced by frogs; and the property CROAK$_2$, which is a property of people who died. The underspecified language proposed by Poesio would allow us to define three different predicates: CROAK$_1$, denoting the set $\{f_1\}$ where f_1 is a function from situations to sets of frog-sounding objects in those situations; CROAK$_2$, denoting the set $\{f_2\}$ where f_2 is a function from situations to objects that die in those situations; and a predicate CROAK$_U$, whose denotation would be the set $\{f_1,f_2\}$. The predicate CROAK$_U$ could then serve as the underspecified $\langle e,t \rangle$ translation for the lexical item *croak.*

Poesio proposes a language that can be used to assign a unique translation to sentence constituents which are scopally, referentially, or lexically ambiguous. The sentence *I don't see the engine*, for example, can be represented as in (3.7); the translation of the definite *the engine* is derived from Heim's proposal [Heim, 1982], as discussed in [Poesio, 1994].

[9] It is assumed that structural ambiguity is resolved by a separate module.

(3.7)



Logical Forms such as (3.7) denote the set of propositions that can be obtained by (i) assigning a scope to all the operators—quantifiers, negation, etc.—by means, say, of a procedure like Cooper's Store [Cooper, 1983]; (ii) choosing a particular value for all *parameters*, of which (3.7) contains two, $\dot{y}$ and $\dot{s}$; and (iii) choosing a specific interpretation for all underspecified lexical items, such as SEE$_U$.

Semantic interpretation is seen as the process of computing the extensions of the theory consisting of the previous context augmented with the underspecified representation of the new sentence under 'discourse interpretation principles' coded as default inference rules. *Perceived Ambiguity* is the state that may occur at the end of the interpretation process if more than one interpretation is available for an utterance in a discourse situation. Perceived ambiguity is therefore defined in terms of the workings of the process of hypothesis generation in context. In this way, the Combinatorial Explosion Paradox is addressed without ruling out the possibility that a human (or a system) may recognize a sentence as ambiguous. Because of the assumption that the process of disambiguation is driven by unsound default inference rules, the notion of 'sound inference' in this framework is not explored in great detail.

An earlier version of the theory in which no disambiguation at all was performed was used as the basis for the TRAINS-92 system, and the version discussed here as the basis for the TRAINS-93 system. The approach used in TRAINS-93 allowed a much simpler interface with the planner and the discourse reasoner. The major problems encountered by this approach were the need to provide a structural analysis of large portions of text, and the interaction between different disambiguation procedures.

Chapter 4

Contextual Reasoning

‘Contextual reasoning’ refers to the inferential processes that are called on in

- interpreting context dependent aspects of a sentence
- selecting the most likely reading of a sentence that is in isolation ambiguous

Some of the linguistic phenomena that require the first type of contextual reasoning are nominal reference, ellipsis and sentence fragments, inferred relations, (possessives, “light” verbs like “have”, “do”, “make” etc.), tense and aspect, implicit arguments (participants in a proposition whose presence is not explicit but which must be inferred for coherence), and various types of metonymy and coercion.

The most obvious disambiguation phenomena are those concerning quantifier scope, sense selection, and structural disambiguation. Some phenomena (pronoun reference, ellipsis resolution) require both types of inference, both to suggest and to select between candidates.

Contextual reasoning is not a highly developed area of natural language processing, despite its importance. There are few implemented systems that carry out such reasoning in any very highly developed way. Thus this section is able to report very few solid achievements, but merely points to a range of techniques that have been used to address some aspects of contextual reasoning, and which may yield results in the future.

4.1 Techniques

4.1.1 Marker passing

Marker passing is a form of network propagation, a computational technique that many people in AI have been interested in for performing simple kinds of inference, since it lends itself very well to special purpose hardware. (Historically, it was a precursor of "connectionism" and related techniques).

In NLP, versions of the marker passing technique have been used chiefly in lexical disambiguation. In the simplest cases, word senses are associated with nodes in an associative network. When a sense hypothesis is considered, a marker is propagated from the node corresponding to it. Hypotheses whose markers intersect with those of other words in the sentence are those which are favoured.

Alshawi ([Alshawi, 1987]) describes a more sophisticated use of marker passing which is also capable of carrying out certain types of reference resolution, and of modelling contextual salience. The book also contains a good overview of the literature.

To date, marker passing approaches have been of limited practical utility, since they require a large amount of effort to be invested in building the requisite knowledge structures. However, it may be that recent work in statistical corpus techniques for word and sense clustering could lead to a resurgence of interest in the area (or perhaps may lead to techniques that turn out to subsume it). Also, the recent availability of large machine readable dictionaries with some thesaurus-like structure may mean that the knowledge acquisition bottleneck can be overcome.

4.1.2 Deduction

Many systems have used forward or backward chaining inference systems as a component of contextual resolution. Inference rules are used to locate candidates for, for example, pronouns or definite descriptions, typically by constructively proving theorems like (for "the Noun"):

```
exists(X,  Noun(X) & not(exists(Y, Noun(Y) & not(X=Y))))
```

An instantiation for X is the referent of the definite description, in standard cases.

Inference rules can be used to construct "inferred antecedents" in cases like this:

```
ABC reported large losses.
The managing director ....
```

The information that companies have managing directors is modelled by an axiom which in skolemised form would be:

```
company(X) -> employ(X,sk0(X)) & managing_director(sk0(X))
```

Given the information that ABC is a company:

```
company(abc)
```

we can then prove that there exists a (contextually) unique X satisfying the description, thereby resolving "the managing director" to "sk0(abc)" which is sufficient to be able to identify the person in question. (It is not too difficult to also produce more human readable equivalents of the skolem functions.)

4.1.3 Abduction

This type of deductive reasoning, with the exception of "closed world" phenomena, presupposes that the relevant knowledge is completely axiomatised (and axiomatised in a computationally tractable form). Except for very limited and simple domains this is a wholly unrealistic assumption. Some knowledge cannot be adequately represented in a deductive way at the current state of the art. Relevant axioms may have been omitted, or cast in an inappropriate form.

One way of overcoming this problem is to use "abduction": that is to say, to reason on the basis that given Q and "if P then Q", then we can assume P if no contradiction arises. Of course, this is not logically valid, but it is a plausible inference in many circumstances. For example, suppose that ABC in the illustration above was a name unknown to an NLP system, and which therefore did not have the information that “company(abc)”. The system would be unable to find a referent for “the managing director” using ordinary deductive inference. However, we know that there is a managing director, and we also know that the only piece of information missing from the deductive inference to this conclusion is the statement that ABC is a company. If making this assumption does not contradict anything else in the knowledge base then an abductive reasoning scheme can make the assumption and thereby determine a reference.

A Prolog-style reasoning system can easily be adapted to carry out this type of inference. In practical implementations the assumptions must be constrained by some kind of cost mechanism which makes it easier to make some assumptions than others. A nice research problem for someone is to determine whether statistical algorithms could automate the assignment of costs to a given set of axioms.

Another type of abductive inference is represented by Charniak’s “abductive unification”. The ordinary first-order unification algorithm is augmented so that it is allowed to match a constant with a skolem function if the result would not cause contradiction. For example, if our text read: “ABC reported large losses. Smith, the managing director, ...” then any reasonable NLP system should be able to infer that Smith is the managing director of ABC, although the text does not actually say this. Assume that the content of the text, and the query as to whether Smith is MD of ABC are represented as follows:

```
company(X) -> employ(X,sk0(X)) & managing_director(sk0(X))
company(abc).
managing_director(Smith).
? employ(abc,smith) & managing_director(smith)
```

Now in order for the query to be answered successfully we need to allow ‘smith’ to match ‘sk(X)’, checking that this produces no contradiction.

Reasonably large implementations have been carried out which use abductive techniques: [Hobbs *et al.*,], [Charniak and Goldman, 1988],

4.1.4 Constraint satisfaction

Constraint satisfaction was first used in computational linguistics by Mellish ([Mellish, 1985]) to solve various reference resolution and quantifier scoping problems. Relevant linguistic and contextual information is collected as a set of constraints and a standard constraint propagation algorithm is used to eliminate candidates until a unique solution is found.

The main difficulty with this approach is that it presupposes that all of the relevant candidates are already accessible. As we have already seen, this is not a realistic assumption.

More recent work within the paradigm of "constraint logic programming" could in principle provide an alternative implementation of the same set of intuitions that drove Mellish's work. The type of constraint simplification algorithms used in this paradigm need not converge on a unique solution if none is available yet, but can produce a description that would be satisfied by any solution. As far as is known, these techniques have not yet been used for contextual reasoning, although they are beginning to be employed in other areas of NLP.

4.1.5 Sortal restrictions

A computationally cheap way of enforcing various constraints is by encoding them as sortal restrictions on arguments to predicates. In a unification based framework, constraints from a sort hierarchy can be compiled into terms and checked by unification ([Mellish, 1988]). Inappropriate combinations of predicate and argument are then filtered out by unification failure, giving a simple approximation to lexical disambiguation.

The technique can also be extended to achieve certain kinds of structural disambiguation, for example by filtering out some modifier attachments if they produce a sortal violation.

Sortal information can also be used to constrain the process of reference or ellipsis resolution, and implicit argument determination. Relaxation of sortal constraints can be used to interpret metonymy or other types of coercions.

Disadvantages of the use of sortal restrictions are: (a) for fine-grained sense disambiguation, it is virtually impossible to arrive at a set of sorts that will give the correct results without also excluding valid interpretations. (b) building a sort hierarchy can be very labour intensive (c) treatment of negation, modal contexts, and conjunction is difficult to achieve efficiently and cleanly.

This is a technique that, like marker passing, can be expected to revive when recast within a statistical framework.

4.2 Alternatives

All of the preceding techniques suffer from the drawback that they require expensive knowledge representation efforts and are computationally intensive and brittle. Most practical systems therefore place reliance on techniques that are less knowledge-intensive and brittle, but which perform reasonably well over a wider range of data.

4.2.1 Structural preference heuristics

A range of preference heuristics have been proposed in the linguistics literature for disambiguating modifier attachments and the like. It is easy to enforce "minimal" or "right attachment" on the basis of the operation of a parser, on the resulting syntactic structures, or even on the structure of resulting meaning representations. To the extent that these heuristics are accurate the results will be better than a random choice most of the time. Analogous preferences for conjunctions have also been used, to prefer the analysis which results in maximally "balanced" conjunctions. Such preferences can be based on syntactic or semantic properties, or a combination of them.

This strategy can be elaborated considerably. For example, a good general heuristic is to prefer interpretations that contain no ellipsis, where possible. Similarly, preferring an idiomatic interpretation to a compositional one where both are possible is usually correct. It is necessary to find some way of calculating an overall score for an interpretation in those cases where several preference measures may apply.

For reference resolution, heuristics based on notions of ''focussing" ([Sidner, 1983]) and "centering" ([Grosz *et al.*, 1983]) are widely used for choosing between a set of available antecedents. Candidate antecedents from earlier parts of the sentence, or preceding sentences, are ranked for salience using structural properties of the sentence, thematic information, and other types of prominence.

Somewhat analogous heuristics have also been proposed for some kinds of ellipsis resolution ([Hardt, 1992]).

4.2.2 Statistical methods

The last few years have seen a surge of interest in statistical methods for many aspects of language processing. One important use of statistical techniques is for gathering data which can be used by some of the inferential or heuristic methods described above: co-occurrence information, sortal restrictions, sense frequencies, etc.

However, statistical techniques can also be used for disambiguation. Sense selection can be performed on the basis of raw frequency, or, given an appropriate corpus to train on, by a form of tagging to prefer high frequency sense collocations.

Attachment and other disambiguation heuristics can be approximated statistically by n-grams of senses, where the senses (i.e. logical constants) are selected on the basis of properties of heads of constituents, perhaps with abstract markers indicating grammatical relationships also being introduced. Thus a low attachment of "I saw the man with a bicycle" might be preferred on the grounds that the score of "man-with-bicycle" was higher that "see-with-bicycle" [Hindle and Rooth, 1983].

Statistical measures can be combined with structural preference heuristics. The ordering induced by individual preferences may be partly or wholly determined by statistical properties. The function that calculates an overall score for those cases where several preferences apply may be statistically trained. In the Core Language Engine, for example, preferences have both an absolute value and a weighting factor. Since some measures may be more relevant and informative in some domains than others, the weights should differ from one application to another. A way of automatically retraining such weights with respect to a new corpus is described by [Alshawi and Carter,].

In the next few years we can expect a good deal of activity and progress in the use of statistical methods to either replace or strengthen existing forms of contextual reasoning.

Chapter 5

Inference and Evaluation

The reason for constructing semantic representations for natural language expressions in the first place is that we want to draw conclusions from linguistic utterances, or want to check the information they convey against a database, or want to judge whether a certain translation of those utterances into another language is correct. Therefore we can say that inference and evaluation are the *raison d' etre* of formal semantic representation.

5.1 Introduction

There is a distinction to be made between (i) reasoning to find out the correct reading of a natural language utterance in a given context and (ii) reasoning to use the information conveyed by the appropriate reading. In what follows we will mainly be concerned with the second kind of reasoning.

Often the logical representations produced by an NPL system give too much information, and we need only part of what is present for database access. The usual procedure in such cases is to first put the logical form representation through a *simplify* program, to strip off what is not needed, and to use the simplified output to access the database.

This squares with the fact that in many cases the full logical representation language of an NLP system is less suited for reasoning. For instance, if the representation language is a version of typed intensional logic, no complete axiomatization exists (for typed intensional logic is not complete), but still, the stripped languages that can be constructed by simplifying typed intensional logic expressions in suitable ways may be much better behaved.

Also, a highly complex representation language L will always include sublanguages which can be got from L by hiding structure (or, as it is sometimes called, by 'zooming out'). For instance, starting from predicate logic one may zoom out to get at the propositional backbone. A theorem prover for propositional logic will be sound for predicate logic as well. It will not be complete, but given a database which represents information propositionally, this will not matter: the theorem prover may well be complete for inference about databases with a suitable level of simplicity.

What this shows is that a natural language representation language need not come with a single theorem prover, but may have a variety of reasoning tools which operate at different levels of granularity. Especially if the representation language is highly complex, reasoning tools for suitable subsets of the full representation language and for suitably simplified versions of it may be very useful.

5.2 What types of reasoning are useful?

5.2.1 Reasoning about valid consequence

The notion of classical inference or valid consequence is given by Tarski's explication as 'transmission of truth':

> Each model of the premisses is also a model of the conclusion.

This can be made more specific by focussing on a particular kind of model and a particular kind of language.

5.2.2 Kinds of models, kinds of languages

In classical propositional logic, the models are $\{1, 0\}$ valuations for a set P of proposition letters, and the language is given by:

formulas $\varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \rightarrow \varphi_2)$.

A proof procedure or calculus for classical propositional logic is a reasoning system which covers valid inference for this kind of model and this kind of language.

A minimum requirement for such a calculus is that it is *sound*: one should not be able to derive consequences that are not valid according to the Tarski explication. A sound proof procedure proves only valid consequences of some language.

The best deductive calculi are also *complete*: any classical inference is derivable in the calculus. A complete proof procedure proves all valid consequences in some language.

A complete calculus exists for classical propositional logic. Moreover, we know that validity for propositional logic is decidable, but that its computational complexity is of the highest kind (NP).

If we move over to a richer notion of model, with a matching language, we can arrive at classical predicate logic. A model M for predicate logic consists of a non-empty domain $\mathrm{dom}(M)$, and a set of relations and operations defined on it, which are the interpretations of predicate symbols and function symbols of a matching language. A function $\mathrm{int}(M)$, the interpretation function of the model M, provides the link. An appropriate language is now given by:

terms $t ::= c \mid v \mid f(t_1 \cdots t_n)$.

formulas $\varphi ::= Pt_1 \cdots t_n \mid t_1 = t_2 \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \rightarrow \varphi_2) \mid \forall v\varphi \mid \exists v\varphi$.

However, the notion of valid consequence in predicate logic is not decidable. A complete calculus for classical predicate logic exists. It gives a procedure for listing all the predicate logical validities, thus proving that the set of valid formulas of predicate logic is semi-decidable. It is impossible to give a procedure which lists all the invalid formulas of predicate logic.

Decidability can be regained if one abstains from using the full power of talking about relational structure: monadic first order logic, which is given by the following definition, is again decidable:

terms $t ::= c \mid v$.

formulas $\varphi ::= Pt \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \rightarrow \varphi_2) \mid \forall v\varphi \mid \exists v\varphi$.

This language can be used for talking about relational structures in a constrained way. We can reason about the property of *being in love with Mary*, but not about the relation of *being in love*.

Starting from propositional logic, we can also move in different directions. We might wish to replace our valuations in $\{0, 1\}$ by valuations in $\{0, 1, *\}$, thus moving from classical propositional logic to partial propositional logic. It is reasonable to extend the language as well. For instance, one might wish to distinguish between weak and strong negation, denoting the latter as $\sim \varphi$. Weak negation acts like classical negation on the values truth and falsity, but maps the value undetermined $(*)$ to itself. Strong negation acts like classical negation on truth and falsity, but maps undetermined to false. Valid consequence for partial propositional logic is decidable. A sound an complete calculus for this logic has complexity NP.

Partial first order logic relates to classical first order logic as partial propositional logic does to classical propositional logic.

Still another direction from propositional logic is to move from classical valuations to sets of propositional valuations (or 'possible worlds'), structured by a two-place relation of accessibility. This gets us into the realm of modal logic. An appropriate language is given by:

formulas $\varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \to \varphi_2) \mid \Diamond\varphi \mid \Box\varphi.$

Here $\Diamond\varphi$ expresses that φ is true in an accessible world, $\Box\varphi$ that φ is true in all accessible worlds. Many modal logics are decidable, and have the same complexity as propositional logic.

Partiality and modality can also be combined, by considering accessibility patterns on sets of partial valuations. See Jaspars [Jaspars, 1994].

Also, it is possible to keep the notion of first order model, but restrict the language, e.g. to the following Horn fragment:

terms $t ::= c \mid v \mid f(t_1 \cdots t_n).$

atoms $A ::= Pt_1 \cdots t_n \mid t_1 = t_2.$

formulas $\varphi ::= \forall v_1 \cdots v_m((A_1 \wedge \cdots \wedge A_n) \to A) \mid \neg\exists v_1 \cdots v_m(A_1 \wedge \cdots \wedge A_n).$

It is assumed that no free variables occur in formulas. This is in fact the language of pure Prolog. A formula of the form

$$\forall v_1 \cdots v_m((A_1 \wedge \cdots \wedge A_n) \rightarrow A)$$

is written in Prolog as

$$A : -A_1, \ldots, A_n$$

and is called a program clause, and a formula of the form

$$\neg \exists v_1 \cdots v_m(A_1 \wedge \cdots \wedge A_n)$$

is written as

$$: -A_1, \ldots, A_n$$

and called a goal clause. The nice thing about this fragment of predicate logic is that any set of program clauses is (by virtue of its form) consistent. Also, there are very efficient procedures to check whether a given goal clause is consistent with a set of program clauses. Prolog is based on one particular choice for such a procedure. The ingredients are : resolution, unification, the Prolog search rule (search the list of program clauses starting from the top), and the Prolog computation rule (reduce the list of subgoals from left to right).

From predicate logic we can also move up, for instance by adding generalized quantifiers:

terms $t ::= c \mid v \mid f(t_1 \cdots t_n)$.

quantifiers $Q ::= \mathit{ALL} \mid \mathit{SOME} \mid \mathit{MOST} \mid \ldots$.

abstracts $A ::= \hat{v}\varphi$.

formulas $\varphi ::= Pt_1 \cdots t_n \mid t_1 = t_2 \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \rightarrow \varphi_2) \mid Q(A, A)$.

Now, a complete axiomatisation may not exist (depending on the choice of quantifiers), but interesting and useful quantifier reasoning tools can be based on the monotonicity properties of the quantifiers.

Another extension of predicate logic is with temporal operators:

terms $t ::= c \mid v \mid f(t_1 \cdots t_n)$.

formulas $\varphi ::= Pt_1 \cdots t_n \mid t_1 = t_2 \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \rightarrow \varphi_2) \mid \forall v\varphi \mid \exists v\varphi \mid P\varphi \mid F\varphi$.

In this case there are complete axiomatisations, but some natural language constructs call for more expressive power, and one might be tempted perhaps to add operators for *until* and *since*. At some point completeness is bound to get lost again.

5.2.3 Reasoning about Particular Models

The most famous example from the history of logic of reasoning about particular models is of course reasoning about the model of the natural numbers, with zero, successor, addition and multiplication. Call this model N. Here we want a calculus which approaches the notion of 'truth in the intended model'. Peano arithmetic is a proposal for such a calculus. However, for the case of arithmetical reasoning we know that no first order calculus satisfies the ideal. Every first order calculus for arithmetic will miss out some arithmetical truths, says Gödels famous incompleteness theorem. In other words: first order arithmetic is incomplete. Stated still otherwise: every set of first order truths of arithmetic will admit non-standard models which look quite different from the standard models. The language of first order logic is simply too poor to nail down the intended model N up to isomorphism.

This problem can be solved by moving to second order predicate logic, which has the following language:

terms $t ::= c \mid v \mid f(t_1 \cdots t_n)$.

formulas $\varphi ::= Pt_1 \cdots t_n \mid Xt_1 \cdots t_n \mid t_1 = t_2 \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid \forall v\varphi \mid \exists v\varphi \mid \forall X\varphi \mid \exists X\varphi$.

In this language it is possible to state the principle of mathematical induction. It becomes (using z as a name for 0 and s as a name for the successor operation):

$$\forall X((Xz \wedge \forall x(Xx \rightarrow Xs(x))) \rightarrow \forall x Xx).$$

Together with the Peano axioms (see any textbook on first-order logic, e.g. Enderton [Enderton, 1972]) this nails down N completely. But at a price. No sound and complete calculus for validity in second order predicate logic exists.

In some sense, Prolog can be viewed as the theory of reasoning about databases of a particular form. This is a huge topic in the model theory of first order logic. For natural language systems, we might also wish to reason about application models of a particular form. For instance, our application models may have the form of Prolog databases.

5.2.4 Proof Strategies for Valid Inference

Proof strategies for valid inference can be subdivided roughly into the following kinds:

- Resolution (plus unification)
- Natural deduction
- Tableau reasoning
- other ...

Fitting [Fitting, 1990] gives more information, plus a very useful introduction to the use of these methods in classical first order theorem proving.

5.2.5 Reasoning About Plausible Consequence

Reasoning about plausible consequence is also called non-monotonic reasoning, defeasible reasoning, preferential reasoning.

Plausible consequence is defined like this (see Shoham [Shoham, 1988]):

> Each most preferred model of the premisses is also a model of the conclusion.

Key question: what is a preferred model? The interesting thing about preference reasoning is that the notion of preferred model may get updated in the course of the reasoning process. See Van Benthem, Van Eijck and Frolova [Benthem *et al.*, February 1993] for a perspective on this process.

5.2.6 Proof Methods for Plausible Consequence

The following is a non-exhaustive list of proof methods for plausible consequence:

- inductive (i.e., case-based)
- abductive (i.e., from conclusion and rule to possible 'cause')
- model based (e.g., model elimination, model checking)
- statistical
- default
- constraint satisfaction.

Plausible reasoning plays a large role in common sense reasoning, so it seems only right that such proof methods are also used in NLP inference and evaluation, but it seems fair to say that there is a bewildering choice of possible approaches.

5.3 Inference and Evaluation in Particular Approaches

5.3.1 Discourse Representation Theory

Some DRS construction rules involve or assume reasoning (see e.g. [Kamp and Roßdeutscher, 1994a]). DRSs can have inference processes defined on them (for the first order fragment see e.g. [Kamp and Reyle, 1991], [Saurer, 1993] and [Reyle and Gabbay, 1994]). Underspecified DRSs are equipped with a proof theory which operates directly on the underspecified representations [Reyle, 1993a]. They may be resolved further by contextual reasoning.

The fully specified language of resolved DRSs has intensional and higher order constructions (e.g., to express propositional attitudes), in short it is a representation language of high complexity to which the above remarks about reasoning with well-behaved subsets and 'zoom out' versions applies.

5.3.2 Dynamic and Update Semantics

The kind of semantic representation used in dynamic and update semantics provides a handle on presuppositions by way of precondition reasoning. If one represents pieces of information as partial relations $\langle R^+, R^- \rangle$, then the presupposition of a piece of information is given as the logical description of the class of contexts (states) s for which either $\{s' \mid sR^+s'\} \neq \emptyset$ or $\{s' \mid sR^-s'\} \neq \emptyset$. See Van Eijck [Eijck, 1994].

There is also a link between preferential reasoning and dynamics. See Van Benthem, Van Eijck and Frolova [Benthem *et al.*, February 1993], and Kameyama [Kameyama, July 1994].

5.3.3 Monotonic Semantics

QLF is input to reasoning to single out the appropriate reading. It has two parts: 'reasoning for resolution' (resolving underspecified relations and establishing anaphoric links) and 'reasoning for disambiguation' (establishing the scopes of operators, most notably quantifiers). 'Reasoning for resolution' is triggered by the presence of particular constructs (possessive pronouns, e.g.). 'Reasoning for disambiguation' takes place via preferences stated on QLFs (stating, e.g., the preference of the determiner *each* to outscope other determiners).

The result of the resolution and disambiguation process is a rich representation language which employs higher order and intensional elements, but which can be 'stripped down' to suit various applications.

5.3.4 Property Theory

Takes an axiomatic approach to the whole subject of natural language semantics. Intensional identity is explicated in terms of 'does the inference system allow the derivation of the identity statement?'

Open question: is there a notion of 'intended model' in PT?

5.3.5 Situation Semantics

This approach aims to use the same framework for reasoning as for linguistic description, which promises a seamless interface between the two kinds of reasoning distinguished in the introduction above.

Clear connections between situation semantics and partial logic are given in the formal reconstruction of situation semantics by Escriba [Escriba, 1992], which also provides a sequent style deduction system for strong consequence in situation semantics.

Chapter 6

Lexical Semantics

6.1 Introduction

Before we turn to more recent developments, it may be helpful to look at some of the lexicon-related problems that present themselves to any generative framework for the description of natural language. We shall do this here from a vantage point that may be unusual in a linguistic context, but which will serve our purpose well enough.

6.2 Categorematic and Syncategorematic Specifications

In formal logic it has been customary to distinguish between “logical” and “non-logical” vocabulary. For instance, a language of first order predicate logic will have a logical vocabulary, consisting of variables, sentence operators, quantifiers and perhaps identity - these it shares with other first order languages - and a set of non-logical constants - predicates, functors and individual constants - which identify it as the particular first order language it is. Intuitively one might expect that each first order language would come with a lexicon, which tells us (i) how the elements of its vocabulary are to be used syntactically - e.g. to what syntactic types or categories they belong - and (ii) what they mean. But as a matter of fact, as first order languages are usually presented, they don’t seem to come with anything for which the term “lexicon” seems at all appropriate. The only component of a standard representation that specifies what could arguably be considered lexical information is the so-called signature of the language, which specifies for each of the non-logical symbols of the language what its logical type is - whether it is an n-place predicate (for some given n) or else an n-place functor.

The reasons why standard presentations of first order languages do not involve lexica less rudimentary than signatures are two-fold. On the one hand the usual, model-theoretic, semantics for first order logic gives a specification of the meaning of the logical constants - the connectives and quantifiers (while the syntactic part of the presentation makes clear how they are used in the building of terms and formulas and thereby reveals what semantic categories they belong to). In such a semantics the meaning of any one of the logical constants is given by the recursive clause of the definition of truth or satisfaction which concerns that constant.

Indeed, it is possible to recast the presentation of first order languages so that it includes a "mini"-lexicon, which has an entry for each logical constant, which carries the same semantic information that is carried by the relevant recursive clause in the "standard" presentations alluded to.[1]

The second reason why many presentations of first order languages supply no more in the way of lexicon than their signature has to do with the difference between "interpreted" and "uninterpreted" logical formalisms. Presentations of the sort we have discussed so far are presentations of "uninterpreted" languages - there is nothing in these presentations that constrains the interpretation of the non-logical constants (beyond the logical types to which the signature of the language assigns them). When a first order language is used to some purpose, however - e.g. to formalize a particular domain, such as, say, a branch of mathematics - then more is required, for now we want to impose on the non-logical constants some particular meanings or interpretations. So it is precisely at this point, one might think, that there is a call for a more extensive lexicon, which specifies the meanings of the non-logical constants of the language. But in fact, the manner in which the meaning problem for interpreted logical languages is normally tackled involves nothing at all like what the linguist would be willing to call a lexicon. Such applications take the form of axiomatic theories - sets of sentences which constrain the meanings of the non-logical constants that occur in them implicitly, by excluding all models in which the constants do not satisfy

[1] This alternative mode of presentation however has the effect of portraying first order languages as sub-systems of higher order logic, something which goes against the deeply entrenched view that the principal metamathematical division runs between first and higher order logic.

those constraints. Only some of the sentences that we find in such axiomatic theories, viz. the *definitions*,[2] can be looked upon as lexical entries (for the non-logical constants they are the definitions of). But in some axiomatic theories there are no definitions at all and in all but a few pathological and degenerate cases there will be at least some axioms that are not definitions.

The reason why this is the way one proceeds in applied logic is plain enough: one cannot specify the meaning of a predicate or functor directly and explicitly unless one has something at hand in terms of which the specification can be given; and for that something to serve the purpose, it should itself have a well-defined meaning already. But if one starts from scratch, as one typically does in such applications, then this is precisely what is missing. In other words, definitions won't do what they are supposed to do (viz specify the meaning of their definienda) if the constants that occur in their definientia do not have meaning on independent grounds. And it appears that the only method for endowing them with such independent meanings is by stating, by means of connecting axioms, semantic relations between them. Otherwise it will be turtles all the way down.

The double moral to be drawn from these observations is that, first, specification of meaning need not take the form of one or more lexical entries and that, second, there are many cases where specification in such a form isn't even possible. Among the methods by which the meaning of an expression can be determined there is on the one hand that of saying how it changes or affects the meanings of other expressions with which it can be syntactically combined - this is the method illustrated by the recursive clauses of truth definitions, in which the items whose meanings are accounted for are treated "syncategorematically" - and on the other the global method of simultaneously constraining several expressions at once through "meaning postulates" which express certain systematic connections between them.

For natural languages the situation is not fundamentally different than it is for the languages of symbolic logic. That lexica and dictionaries nevertheless play such an important role in our commerce with the languages we use has two reasons. First, it is generally true of natural languages that they contain substantial numbers of words which can be defined (or at least approximately defined) in terms of other words of the language. Secondly, ordinary lexica are often circular (i.e. the transitive closure of the relation which holds between words A and B iff B is used in the lexical definition of A is not well-founded). From a practical perspective this is not necessarily a defect, for a lexicon must

[2] E.g.: $\forall x(even(x) \leftrightarrow \exists y(x = 2 \times y))$

offer something to many different users, and some of those may be helped with a definition of A which uses B, while others will be served better by a definition of B which contains A. But this does not change the fundamental and often noted fact that a systematic, non-circular lexicon will, after all possible defining has been done, still be left with a substantial stock of "primitives" - these may either be items of the language itself or else terms of art, which express concepts that are central in lexical semantic analysis without being directly realized by constituents of the language; and the meanings of these primitives can only be elucidated by means of a "lexical theory", which states the connections between them.

At the same time there is at least one respect in which the meaning problem for natural languages is more complicated than our sketch of the meaning problem for languages of first order logic reveals. The complication we are thinking of is connected with the division into logical and non-logical concepts. In the languages of predicate logic this division is treated as absolute: a notion is either wholly logical or else wholly non-logical and, as we have seen, the standard presentations of such languages treat the semantics of these two types of notions in very different ways. However, the absolute division into logical and non-logical notions has been criticized from a number of different angles. Most notorious perhaps are the philosophical criticisms by, first and foremost, Quine, which raise doubts not only for this distinction, but in fact for the whole enterprise of a semantics rooted in an essentially correspondence-based concept of truth, which led to, and still provides the simplest and clearest justification for, the model-theoretic method in the semantics of formal and natural languages. This is a dispute which we will not pursue here. However, the rigid division between logical and non-logical concepts can also be subjected to criticisms of a more homely, but also more tangible sort. These criticisms relate to the often noted fact that the division becomes problematic already when we move from the languages of pure predicate logic which we discussed above to the extensions of them that have been developed within modal or tense logic. The operators that distinguish the languages of tense and modal logic from the underlying languages of pure predicate logic appear to be curious mixtures of the logical and the non-logical functors that we find in the languages of predicate logic with which we started out. Take the tense operators P ("it was the case that") and F ("it will be the case that"), for instance. On the one hand these are treated like the sentence operators of classical predicate logic (syntactically they function as 1-place sentence connectives, such as negation). In particular, the semantic contributions which these operators make are typically captured by the same kind of recursive truth condition clauses that are used to state the semantics of, say, $\wedge$ or $\forall$. But on the other hand the operators also seem to have some features of the non-logical constants, inasmuch as time is, for all we know, no absolute notion, but a concept that might have been realized in some other way than it actually is: in other possible worlds time might be finite, say, or discrete, even if we are right in thinking that in the real world it is infinite and dense. Insofar as the tense operators "refer" to the time that is part of the world or worlds that the sentences containing them are about, the meaning of tense has an aspect of contingency which is typical of non-logical, as opposed to logical notions.

In the standard presentations of tense logics the "hybrid status" of the tense operators is manifest: On the one hand their contribution to semantic evaluation in individual models is handled just as it is for the operators of classical logic (i.e. by means of recursive clauses in the definition of truth or satisfaction); on the other hand their contingency is visible in the variation of time structures that can be found within the class of possible models which provides the link between truth and "logical" validity. Thus the "meanings" which such a presentation associates with the tense operators are composed of two quite different components of the over-all system - the truth definition clause directly corresponding to the operator and the class of models representing possible structures of time.

The tense operators were originally conceived as formalizations of the tenses one finds in natural languages such as English, and so the treatment they receive in the model theories of tense logics might suggest that a similar "two-pronged" strategy is the right approach towards the meaning of tense within a semantics for natural language. Indeed, this is the line that has been pursued by treatments of tense within Montague Grammar. The strategy pursued in DRT is similar, in that it too is two-pronged, but it differs from that of Montague Grammar in that the recursive clauses in a definition of truth are now replaced by construction rules which spell out, for each tense, how the material of a clause with this tense is to be integrated into the given context of interpretation.

The second component of the meanings of the tenses is then provided by the model theory for the DRSs which the construction algorithm (which includes the just mentioned construction rules for the tenses) compute from NL inputs. This second component takes essentially the same form as it does within the setting of tense logics (or, for that matter, in the setting of Montague Grammar), except that the existing DRT treatments of tense and aspect differ from those - implicit in tense logic and explicit in most tense-and-aspect analyses that have been proposed within Montague Grammar - in that they analyze tensed sentences, in the spirit of Davidson, Parsons and others, as descriptions of states or events; but this is a difference which does not affect the points that are at issue here.

These considerations lead us to the following moral. On the one hand there is no doubt that the tenses make quite specific, and identifiable, contributions to the meanings of the sentences in which they occur; they function as syntactically minimal meaning contributors and thus as the sorts of items for which one might expect separate entries in the lexicon of the language. However, in the light of the analyses which tenses and aspect markers have received in the DRT-literature and the linguistic data that these analyses respond to, it seems unlikely that their semantic contributions can be cast in a familiar lexical format. For these operators, DRT would claim, the combination of construction rule and general model-theoretic constraints is as close to a "lexical specification" as we are likely to get. In particular, the syncategorematic aspects of meaning, which, we have argued, have traditionally gone hand in hand with the notion of specification of meaning by recursive rule, and which in classical DRT take the form of specification by means of construction rules, remain appropriate for such expressions in spite of the fact that their semantics has an important contingent component too.

If this is the right conclusion for the markers of tense and aspect, we may expect it to be true of a great many more expressions that one finds in English and many other (indeed, we conjecture: in all other) natural languages, such as: deontic, epistemic, bouletic and other modalities; indexicals such as *I, you, now, here, tomorrow, elsewhere, ..* ; rhetoric particles such as *but, thus, however, moreover, since, because, although, ...* ; discourse-oriented presupposition triggers such as *also, even, other, back, again, ...* , to name but some of the relevant categories and some of the items belonging to them. Indeed, for each of the expressions mentioned here, it becomes clear after even a moderate amount of reflection that specifying its contribution to sentence and discourse meaning requires the same kind of spelling out of "syncategorematic contribution" which has by now become widely accepted in the case of aspect and tense.

The upshot of this is that if we think of lexical information as that which is specific about the semantics of the minimal meaningful units of a natural language, then a lexicon which fits the common conception of what lexica are or ought to be like will cover only part of it. In addition to a lexicon as a collection of lexical entries which provide definitions for their lexical items, we will need many lexical specifications of the syncategorematic sort; moreover, as observed earlier, we will need a substantial "lexical theory" to state the semantic connections between the semantic primitives to which the lexical definitions ultimately recur.

6.3 Lexical and Computational Semantics

After this cautionary tale about the limits of lexica in familiar form (i.e. lexica which consist of collections of lexical entries providing definitions, or at least approximate definitions for lexical items) let us now turn to the problem of lexical entries for items that do seem to permit lexical specification along roughly traditional lines.

Computational Semantics of the kind we pursue within the framework of FraCaS is rooted in "Formal Semantics", the discipline which grew out of the work of Montague and other logicians and logically oriented linguists and philosophers in the sixties. But Formal Semantics has, in the thirty years or so that it has been around, paid comparatively little attention to the semantics of individual words. In retrospect this is no less understandable and justifiable than it appeared at various points in the past when formal semanticists made conscious or semi-conscious decisions about the problems they should work on. For without the fairly systematic understanding of the compositional mechanisms that are involved in determining the meanings of complex linguistic structures such as sentences and texts, it would not have been possible to treat individual word meanings with the rigour and precision which constitute the formal semanticist's methodological sine qua non and which are indispensible where linguistics is to serve as the basis for computational linguistics.

These observations about the recent history of natural language semantics should not blind us, however, to the fact that it *is* lexical semantics which constitutes the bulk of what is needed in an exhaustive characterization of meaning for any natural language - the sort of characterization, in other words, without which we will never be able to build the sophisticated computational applications that are expected by the world at large. There is an enormous amount of work waiting to be accomplished here, the size of which exceeds what has been accomplished within Formal Semantics so far by an order of magnitude.

To avoid misunderstanding: This does not mean that there is no significant body of results in computational work on the lexicon - as a matter of fact there already exist large coverage lexica for a number of languages, which are used in syntactic parsing, enabling the parser to represent, as part of the parses it produces, the predicate-argument structure of clauses and other complex phrases. For many of the operations that a sophisticated application must be able to perform on semantic representations - in particular those concerned with monotonic and non-monotonic inference - this information is not enough, however.

Conceived broadly work in lexical semantics (this includes both computational and theoretical work that is relevant to computational semantics) has dealt with (at least) the following - often interrelated - issues:

- how to encode lexical semantic information with the least amount of redundancy (ambiguity, underspecification, qualia structure, etc.)
- how to relate lexical semantic information both syntagmatically and paradigmatically (word fields, clustering, collocations etc.)
- how to "explain" the meaning of words (decomposition, quest for semantic primitives, conceptual structures, etc.)
- how to relate syntactic with "semantic" arguments (LMT (lexical mapping theories), linking theories, conceptual structures, thematic roles and grammatical functions, etc.)
- how to account for "lexically triggered" inferences, (meaning postulates etc.)
- formalisms and tools (lexical redundancy rules, (multiple) inheritance hierarchies, taxonomic specifications, defaults, monotonicity, etc.)
- how to compile computational lexica (and the semantic information encoded therein) from machine readable lexica (lexicon construction in general, corpus work etc.)

Compared to the sheer size of the task at hand it is not surprising that there is comparatively little by way of extant work on lexical semantics which is *directly* usable to computational semantics as we understand it. There are important beginnings, such as the work by Briscoe, Copestake (c.f. [Briscoe *et al.*, 1993], [Copestake, 1993], [Copestake, 1990]), Boguraev [Boguraev and Briscoe, 1988] and others, the work by Pustejovsky (c.f. [Pustejowsky, 1991]), that by Bierwisch (c.f. [Bierwisch, 1983]), Lang (c.f. [Lang, 1985]), Wunderlich [Wunderlich, 1987] and other researchers. The formally less explicit, but highly preceptive work of people form the MIT Lexicon Project, such as Hale (c.f. [Hale and Keyser, 1986], [Hale and Keyser, 1987]), Levin (e.g. [Levine, 1993]) and

Rappaport (c.f. [Rappaport and Levine, 1986]), should also be mentioned here. In addition there exists much work that has been done in the AI community (e.g. Schank's work on semantic "primitives" in the framework of conceptual dependency theory, c.f. [Schank, 1972]) and while a good part of this does not seem suited to computational semantics as conceived here, some of it surely does. Finally, a start has been made on the formal semantics of lexical items within some of the frameworks represented within FraCaS. Here we will just give some pointers to DRT-based work, on the one hand that which relates to the theory of Segmented DRSs of Asher (c.f. [Asher, 1993]), which has close connections with the work by Briscoe and others which we mentioned above, and on the other that by Roßdeutscher ([Roßdeutscher, 1994], [Kamp and Roßdeutscher, 1992], [Kamp and Roßdeutscher, 1994a] and [Kamp and Roßdeutscher, 1994b]).

Chapter 7

Survey of Implementations

This section contains a survey of implemented semantic formalisms. We look at those NLP systems which contain interesting semantic components. For each system, we will mention its background and intended function, describe which semantic theory or formalism is used, and indicate the coverage of the system, including the classes of semantic phenomena treated. It would be very instructive to compare the coverage of these systems with respect to the criteria developed in the FraCaS project, i.e. the spectrum of selected phenomena and the computationally relevant aspects discussed in chapter 9 of the present document. However, we have not done it for several reasons. For one thing, it is very difficult to judge the performance of systems from the outside, as it were. Usually, only unsufficiently documented prototype versions exist, which are not freely available, and it is very difficult to assess system performance on the basis of papers, casual documentation, and system demonstrations. It is also close to impossible to extract reliable information from the system builders by questionaires or interviews: The choice of certain features and phenomena for implementation often depends on specific requirements of the domain or on preferences of the people involved rather than what is supported by or compatible with the underlying framework. Furthermore, the information that a certain phenomenon is covered may mean many different things: It may be just represented on the notational level for the purpose of semantic composition; it may be denotationally interpreted; inference techniques may be available for (some or all) contextual resolution tasks connected with the phenomenon, or it may be the case that the system provides reasoning techniques to exploit the specific kind of information connected with the phenomena.

Thus, we will not provide a detailed evaluation of all systems. For each system, we will give a short description, which mentions its background and intended function, describes which semantic theory or formalism is used, and indicates the coverage of the system. In the folloowing chapter, we will look at two of these systems, i.e., CLE and Verbmobil, in more detail, trying to make a bit clearer to which degree these systems meet or can be expected to meet the criteria.

7.1 The Core Language Engine (CLE)

The CLE has been developed at SRI Cambridge over the course of the last 7–8 years. It is a 'sentence in, logical form out' system, employing a pipe-lined architecture with modules covering: segmentation, morphology, phrasal parsing, full parsing, semantic analysis, generation, reference resolution, resolution paraphrasing, scoping, translation of logical forms into domain specific representations. Although the system is primarily rule-based, statistically trained preferences can be applied at various points to filter out unwanted analyses.

A broad coverage, unification-based syntactic and semantic grammar of English has been developed for the CLE (approx 70% of 1–10 word sentences in LOB assigned a plausible semantic analysis/QLF), and the grammar has been adapted for other languages, e.g. Swedish and French. The grammar is fully reversible, and is used for both analysis and generation.

Semantically, a fair indication of the CLE's coverage is given by the description of QLF in deliverable D8. It includes: singular and plural quantification, anaphora, ellipsis, comparatives, tense, compound nominals, adverbial and adjectival modification. Analysis to the level of unresolved QLFs is reversible. Reference resolution is partially reversible, allowing generation of more explicit paraphrases of resolutions. Reference resolution allows for calls to be made to general domain level inference, which has been used e.g. to assess the domain plausibility of quantifier scopings or suggest domain-specific interpretations of compound nouns. However, domain specific resolution and inference remains one of the hardest areas to deal with. Unresolved QLFs have been employed in machine translation systems based on semantic transfer.

7.2 $\mathcal{NLL}$

The logical language of $\mathcal{NLL}$ (natural language logic) was designed at Hewlett-Packard labs and the German Research Centre for Artificial Intelligence (DFKI). It aims to provide tools for the construction of a semantic structure from syntactic and lexical structures and a formal base for disambiguation and domain-specific interpretation, which, however, is itself not specified within $\mathcal{NLL}$.

The fundamental aim was to provide a logical language for natural language processing without committing the system to any one particular semantic theory, let alone suggesting a new theory. $\mathcal{NLL}$ defines a core which does not include much more than the features of standard predicate logic. The core does not adopt a particular view of the special topics currently discussed in natural language semantics, such as tense, temporal expressions, events and propositional attitudes. However, it is designed in order to experiment with all these extensions, according to the different current theories. It is therefore intended that $\mathcal{NLL}$ does not define a particular interpretation or supplies a processing component; these are meant to be fixed by the application using $\mathcal{NLL}$.

$\mathcal{NLL}$ provides a model theory for the core of the language, and also suggests how to add definitions for some of the extensions. There are also extensions which are not interpreted. Meaning postulates define the intended meaning of operators for the user, but there is no component within $\mathcal{NLL}$ which forces the suggested meaning; this must be defined when putting $\mathcal{NLL}$ to use.

In an example application, $\mathcal{NLL}$ structures were built from the feature structure output of a syntactic processing system which were then passed on to a pragmatics module. Specific semantic phenomena that have been treated within $\mathcal{NLL}$ include quantification and plurals.

7.3 Rosetta

The automatic translation system Rosetta [Rosetta, 1994] uses the concept of isomorphic Montague grammars to translate between languages like English, Spanish and Dutch. Key notions are 'translation equivalent basic expressions' and 'translation equivalent rules'. The isomorphism relation between the grammar for English and the grammar for Dutch are set up by stipulating which basic expressions and which rules are translation equivalent. For example:

$\mathrm{N}(\mathit{girl}) \equiv \mathrm{N}(\mathit{meisje})$

and:

$R_{\mathrm{English}}{}^{n}$: $\mathrm{IV}(\alpha) \Rightarrow \mathrm{VP}(\mathit{does\ not}\ \alpha)$
$R_{\mathrm{Dutch}}{}^{m}$: $\mathrm{IV}(\alpha) \Rightarrow \mathrm{VP}(\alpha\ \mathit{niet})$
$R_{\mathrm{English}}{}^{n} \equiv R_{\mathrm{Dutch}}{}^{m}$

Of course, once two grammars are linked in this manner, it is possible to define a 'shared' semantics for the two languages defined by the grammars, by giving equivalent basic expressions the same basic meaning, and linking equivalent syntactic rules to the same meaning rule. In this way, expressions of Montague's intensional logic may serve as an *interlingua* between source and target language in the translation. It should be noted, though, that the route via Montagovian meanings is not essential: the isomorphic structure trees are enough to establish the connection. Chapter 2 of [Rosetta, 1994] gives more information.

7.4 Squirell

SQUIRREL is a portable natural language front end that uses a formal, rule-based approach to syntactic and semantic analysis [De Roeck *et al.*, 1991c; De Roeck *et al.*, 1991d]. The original application produces queries (in SQL) that are presented to a relational database (INGRES). A version of SQUIRREL has been used to provide input to a nonmonotonic theorem prover [De Roeck *et al.*, 1991a].

SQUIRREL uses a context free feature-value grammar and a bidirectional chart parser [Steel and De Roeck, 1987]. During parsing, a semantic representation is built which consists of a property-theoretic term. After parsing, for each successful analysis, β-reduction is performed on the semantic representation to produce a normal form, and the result is checked for propositionhood. Assuming the representation is a proposition, the axioms of truth are effectively applied by way of a recursive definition expressed in Horn clauses. This yields a representation in first order logic. *Wh*-categories are represented using free variables.

To eliminate domain dependence in the lexicon, the grammar attributes appropriate generic attribute values to "unknown" words. When attached to the database back end, the first-order representation is mapped into an SQL query by way of various relational calculii. The initial translational step requires the use of domain specific information, which is isolated in an *Extended Data Model* (and in a *Sortal Hierarchy* for the theorem proving back end). Various meaning-preserving optimisations are applied to the query during the course of translation. Nonsensical queries are blocked by a "type" checker. After type checking, the final query (or queries) can be paraphrased back into English [Lowden *et al.*, 1993a] before being presented to the database.

Using a rule-based approach with formal semantics means that the translation to SQL is systematic and transparent. This helps to eliminate the errors that can result from less rigorous techniques, where the query presented to the database may not correspond with the users intentions. It also simplifies the process of extending the front-end to cover additional phenomena, such as anaphoric, and modal queries [Barros and De Roeck, 1994; Lowden *et al.*, 1993b].

7.5 Tacitus

Tacitus is a general purpose NLP system developed by Jerry Hobbs and his group at SRI International's Artificial Intelligence Center, Menlo Park. It is implemented in Lisp, and uses an augmented context-free grammar formalism descended from the 'Dialogic' system building essentially first order 'ontologically promiscuous' logical forms ([Hobbs, 1991]). 'Ontological promiscuity' in this context means that elements of linguistic structure are related to abstract or virtual elements which may or may not correspond to elements in the real world. Contextual resolution is the process of relating the virtual to the real objects, adding extra information as a side-effect of the necessary inferential processes.

The interest of the Tacitus system lies less in its linguistic processing than in the fact that it tries to tackle the contextual reasoning problem in a general and motivated way. Contextually dependent constructs are represented by particular dummy predicates in logical form. These are treated as goals to be solved with respect to a database containing general and contextual information, and the content of the rest of the sentence. The inference method used is 'weighted abduction' ([Hobbs *et al.*,]): an essentially Prolog style backward-chaining proof procedure is augmented with an 'assumption' step, allowing certain goals to be assumed if they cannot be proved, provided no contradiction arises. Each assumption has a cost, so that inferences with no, or few assumptions are preferred. A further type of abductive inference is 'factoring' of unifiable subgoals: factoring unifies the subgoals, again if no contradiction arises. This technique is used for interpreting pronouns, missing arguments, resolution of compound nominals, metonymy, etc.

This style of contextual resolution requires an explicit logical axiomatisation of the structure and properties of the domain of application. Even with as few as 400 axioms, the computational load can be heavy ([Appelt and Hobbs, 1990]). As far as is known, the Tacitus system represents the most ambitious attempt to tackle the contextual reasoning problem using 'real' reasoning.

7.6 Trains

The TRAINS project (1990-) is an effort to build a conversationally proficient planning assistant under way at the University of Rochester, Department of Computer Science. A key part of the project is the development of prototype systems which provide the research platform for a wide range of issues in natural language understanding, mixed-initiative planning systems, and representing and reasoning about time, plans and events. The TRAINS system helps a user construct and monitor plans about a railroad freight system; their interaction takes place in natural language. A demonstration system has been developed each year since the inception of the project; these systems are able to replicate a conversation selected from a corpus of transcribed spoken conversations between two humans, one of which playing the role of 'system', the other the role of 'manager', collected at the University of Rochester.

Although several different representation languages are used by the modules of the TRAINS system—each tailored to a particular task—a considerable effort went into studying the semantic relation between these formalisms so as to ensure correct translations. The logic used for (lexical) semantic interpretation and parsing, Episodic Logic (EL), is the most developed among these formalisms. EL is a first-order, situation-based logic which incorporates ideas from many existing semantic theories, and has been designed to support a Montague-like coupling between syntactic interpretation and semantic interpretation. As its name says, an important aspect of EL is the inclusion in the language of the ability to explicitly refer to *episodes*, which are situation-like objects, since the primary use of EL has been in dealing with temporal reference, aspect, and event anaphora [Hwang, 1992]. The grammar used in TRAINS covers a subset of English that overlaps to a large extent with the fragment discussed in Deliverable 8. Episodic Logic comes with an interpretation component called EPILOG, which has not, however, been used for reasoning tasks in the TRAINS prototypes developed so far.

The result of semantic interpretation in TRAINS-93 (the most recent prototype system) is an Underspecified Logical Form, which is used as the starting point for 'surface discourse interpretation', or *deindexing*, which includes reference resolution, the interpretation of temporal anaphoric expressions, and scoping. This is implemented as a process of inference over expressions of a language called CRT [Poesio, 1994], that extends Episodic Logic syntactically (the most notable addition is the inclusion of DRS structures for explicit reasoning over discourse referents) as well as semantically (by 'generalizing' the definition of the interpretation function so as to provide a semantic characterization of the meaning of underspecified representations). The result of deindexing is a non-indexical, scoped logical form that can be given a semantics within Episodic Logic proper.

The representation language used by the Dialogue Manager, called EBTL, is fairly close to Episodic Logic as well, and it comes with a (partial) axiomatisation that has been implemented on top of the RHET knowledge representation system developed at the University of Rochester, that includes facilities for forward and backward reasoning together with specialized reasoners for type inference and temporal consistency checking. Finally, a fourth specialized representation is used by the Plan Reasoner for plan consistency, plan recognition, and plan generation inferences.

7.7 Verbmobil

The Verbmobil project (1993-) combines speech technology with machine translation techniques in order to develop a system for translation in face-to-face dialogues. The Verbmobil system will provide English translation for negotiation of business appointments between German and Japanese users who have only a passive knowledge of English. The major requirement is to provide translation as and when users need it, and do so in real-time. In order to meet this requirement, the system is composed of time-limited processing components which perform acoustic, syntactic, semantic and pragmatic analysis, transfer, dialogue management as well as generation and synthesis.

The semantic formalism being developed in the Verbmobil project, λ-DRT, combines the basic features of DRT with Montague-style Extended Type Theory to obtain compositionality. In essence, DRSs (pairs consisting of a set of discourse marker and a set of conditions on these marker) are taken as the basic meaning expressions but λ-abstraction over DRSs allows the construction of complex meaning expressions. Semantic representations are constructed bottom-up from HPSG-style syntactic structures. The formalism has also been extended to incorporate: an underspecified treatment of quantifier scoping, anaphora and ellipsis; a neo-Davidsonian treatment of adjuncts; deictic referents; thematic roles and conceptual types defined in an external domain model; and links to instantiated concepts in a contextual discourse model. The transfer module uses this semantic information, together with pragmatic information, to map between source and target language utterances. The formalism has been implemented in a typed feature description language (STUF-III) and incorporated in a 'mini' demonstration system for a fragment of spoken German.

The current linguistic coverage of the formalism includes: quantifiers, nominal anaphora, ellipsis, adjectives, temporal reference, verbs, questions and event-type anaphor (partial). Plurals, comparatives and attitude verbs will be treated in an extended version of the formalism.

Chapter 8

Evaluation of Two Systems

The following chapter tries compare two important implemented systems with respect to coverage and other relevant properties, in somewhat more detail.[1]

The two systems commented on in the following, CLE and Verbmobil, differ in at least two important respects: First, the CLE is intended to be a multi-purpose linguistic core machine, where Verbmobil is dedicated to a special purpose, namely spoken dialogue translation. Second, the CLE has a history of about 8 years, whereas Verbmobil only started in summer 1993.

8.1 The Core Language Engine

The system referred to in the following is that at the end of the CLARE project (December 1992). Subsequent work on the CLE has been geared towards spoken language translation systems, with an emphasis on (a) extending coverage for English to unresolved QLFs, (b) adapting the English grammar and semantics to provide descriptions for other languages, (c) the use of statistical techniques for adapting the system to specific domains (e.g. ATIS, hotel bookings). Outside of an interface to an automatic route finder, comparatively little work since 1992 has been devoted to contextual reasoning and resolution.

[1] At the same time, it illustrates the difficulty of making useful statements about the evaluation of systems (c.f. the introductory remarks in the preceding chapter).

8.1.1 Phenomena

8.1.2 Generalized Quantifiers and Scope

All kinds of GQs can be represented, though the analysis of highly context dependent quantifiers (e.g. *many, few*), is schematic rather than complete.

For scope resolution, both syntactic information, and domain reasoning about (absolute) functional relationships between objects are used to disambiguate scope. Scoping is performed through instantiation of scope constraints at various points throughout a QLF, and does not directly implement Cooper storage.

Most forms of constituent coordination are dealt with (including NP, S, VP, V, Nbar, Det, PP, Adjp).

8.1.2.1 Plurals

Plural entities and anaphora can be represented through the use of 'set quantifiers'. There is a treatment of collectives and distributives, but it would benefit from further development.

8.1.2.2 Anaphora

Inter and intra-sentential anaphora, pronominal and definite description anaphora, plural, E-type donkey and proper name anaphora are treated, through the selection of contextual restrictions on the domain of quantification of anaphoric terms. Functional anaphora can also be handled, though the domain specific coverage is limited.

8.1.2.3 Ellipsis

Various forms of VP, NP, PP, Nbar and Determiner ellipsis are covered, both inter- and intra-sententially. Ellipsis for answers to questions falls out from the general treatment of ellipsis.

8.1.2.4 Adjectives

Compositional treatment of all kinds of adjectives, though only intersective and ordinal (i.e. 'first', 'last') adjectives support reasoning in a fully coherent way.

8.1.2.5 Comparatives

A wide variety of comparative constructions are dealt with, including elliptical comparative constructions.

8.1.2.6 Temporal Reference

A basic treatment of tense and temporal reference is provided at the level of contextual resolution. While in need of improvement for strongly temporal domains, it goes a surprisingly long way in handling domains where time is not the central concern. Topics covered: simple treatment of past, present, future, perfect temporal reference; over simple treatment of the progressive; (quantificational) temporal adverbials; temporal indexicals; partial treatment of temporal connectives. Aspectual distinctions (mainly state vs event) given a basic treatment in a general domain model.

8.1.2.7 Verbs

Aspectual properties of verbs dealt with in a domain model, determining how QLFs should be translated to equivalent domain specific logical expressions: aspect only relevant here if it affects translation (e.g. temporal PP "on" indicates overlap or containment when modifying states or events). Otherwise, little treatment of aspect.

Resolution of light verbs (copulas, "have", etc) supported. Modals covered in semantic representations, but little has been done about their model-theoretic interpretation. De dicto and de re ambiguities handled as part of quantifier scoping (intensional verbs can be given distinct scoping preferences from non-intensional verbs), but model-theoretic interpretation left open as with modals.

8.1.2.8 Attitudes

No sensible treatment of attitudes. But can handle performative verbs, e.g. displaying information in response to requests to "show me ...". Also domain modelling of meta-knowledge through reasoning about when and where the closed-world assumption applies, e.g. "do you know who works on the CLARE project?".

8.1.2.9 Questions

Sentence mood indicated in QLF. Multiple Wh-questions supported, as are many indirect questions. Pragmatic, domain specific reasoning can sometimes determine when a yes-no question requires more than a yes-no answer: e.g. "do you know who...".

8.1.2.10 Events and Event-Type Anaphora

QLFs currently resolved to give Davidsonian event-based treatment, offering potential for event anaphora; but event anaphora has not been implemented, partly due to doubts about how applicable the Davidsonian treatment really is.

8.1.3 Computationally Relevant Aspects

8.1.3.1 Interface to Syntax

A fairly standard, unification-based syntax-semantics interface is employed to build up unresolved QLFs. While it would be possible to interleave syntactic and at least the initial stages of of semantic processing, a sequential, pipe-lined architecture is employed for efficiency reasons. (Checking semantic plausibility tends to be expensive, and the expense is needless if a constituent is later rejected on syntactic grounds). Statistically trained preference metrics are used to select candidate QLFs in the case of syntactic ambiguity. A semantic-head driven generator provides a reverse, semantics-syntax interface, using the same syntactic and semantics rules under a different compilation regime.

8.1.3.2 Underspecification

Semantic ambiguity and vagueness is represented in underspecified form through the use of uninstantiated meta-variable in the semantic representation. Resolution leads to the instantiation of the meta-variables to appropriate values. The order in which meta-variables are instantiated is unimportant, permitting a variety of processing regimes for contextual resolution.

No attempt has been made to combine syntactic and semantic/contextual underspecification within one formalism. For speech input, where there may be many candidate syntactic hypothesis, processing is done on an N-best basis: the N best syntactic candidates are independently processed and their semantic plausibilities compared.

8.1.3.3 Contextual Reasoning

A domain model can be specified detailing how resolved QLF expressions may be translated to logically equivalent expressions in some domain specific notation (e.g. SQL queries). This translation and other material from the domain model mediates the contextual reasoning used in resolving underspecification. For example, functional information contained in the domain about possible relations between objects is used to constraint quantifier scope resolution; common domain relations between objects are used to suggest possible resolutions for compound nouns and prepositional relations. Contextual reasoning is also used to determine whether contextually salient objects satisfy certain QLF predications, and this is used heavily for anaphora resolution (especially definites).

The context model (lists of recently mentioned entities, QLFs for previous sentences etc) is maintained separately from the domain model proper, though contextual reasoning can make use of both sources of information.

8.1.3.4 Inference and Evaluation

Inference and evaluation is not carried out on unresolved QLFs, nor even directly on fully resolved QLFs. Instead, an equivalence preserving translation onto a target reasoning language is performed via a domain axiomatisation. The translation procedure is itself inferential, and there is no absolute distinction between the inference done for QLF translation and QLF evaluation.

8.1.3.5 Lexical Semantics

The CLE has a core lexicon of some 2500 entries. It can be run with general purpose lexica (e.g. the MRC lexical database), though this typically gives coarser classification of the syntactic and semantic properties of words. The system provides semantics for inflectional and a certain amount of derivational morphology. For certain classes of closed-class words, such as prepositions, no attempt at providing different words senses is given. This is because the interpretation of such expressions can be highly domain specific — arguably, prepositions do not have discrete word senses. A sortal hierarchy is provided, though the work of this has largely been supplanted by the use of statistically trained, domain specific word collocations.

8.2 Verbmobil

The system version referred to in the following is that of the VM Demonstrator, which is in the final software integration state now (public presentation in February 1995). At several points, future perspectives (viz., the VM Reserch Prototype planned for the end of 1996) are also referred to.

8.2.1 Phenomena

Since the project is emperically driven, the system is limited to the required coverage and functionality. Being a speech system, the limitation to the phenomena the speech component can deal with is severe and leads to a different coverage than D2.

8.2.1.1 Generalized Quantifiers and Scope

All kinds of GQs can be represented. There have been no attempts to interpret more complex cases of quantifiers, nor to implement reasoning techniques for their scope resolution, since they are not the type of expressions found in the Verbmobil corpora (appointment making domain).

Quantifier scoping is done by a Cooper Storage technique. Using the interface facilities, different kinds of syntactic information can be easily accessed from the semantics to constrain scoping possibilities.

S- and simple NP coordination can be treated. No systematic treatment of more complex cases, since no support by syntax so far.

8.2.1.2 Plurals

Pluralic entities and plural anaphora can be represented. No collective/distributive analysis so far.

8.2.1.3 Anaphora

Inter- and intra-sentential anaphora, pronominal and definite description anaphora, plural, donkey, and functional anaphora are treated in a DRT-style framework, using an extension of the theory of anaphoric binding, accommodation, and presupposition of van der Sandt.

8.2.1.4 Ellipsis

Implementation of syntactic and discourse ellipsis is under work, one of the main themes for the second (Prototype) phase; no implemented version so far.

8.2.1.5 Adjectives

Compositional treatment of all kinds of adjectives; only intersective ones get interpreted representations so far.

8.2.1.6 Comparatives

Simple cases of comparatives and superlatives can be treated, e.g.

> *Er spielt besser.*

There is a comprehensive λ-DRT account of comparatives, however, which has been independently implemented and could be easily integrated.

8.2.1.7 Temporal reference

Standard use of tenses, indexical adverbials, temporal clauses (*before*, *after*), quantificational temporal adverbials, temporal adverbs of quantification are in the implemented fragment. For temporal PPs likewise; the modelling of their effect on verb aspect is under way. Work on temporal anaphora is also in progress. It is implemented in terms of a sort/conceptual type specification which is outside the interpreted part of the formalism.

8.2.1.8 Verbs

Detailed account of aspectual verb classes finished for demonstrator. No de re - de dicto treatment.

Copula with different alternative readings is in the implemented fragment. Modals,including complex modal constructions, are very frequent and therefore provided with a detailed semantic representations, but with a domain-specific and quite coarse interpretation.

8.2.1.9 Attitudes

No treatment of attitude verbs in the narrower sense, but of performative verbs and constructions, which are as frequent and diverse as modals in the domain, but express but a few standard performative acts. The analysis is pragmatic, no specific semantic interpretation.

8.2.1.10 Questions

Sentence mode is represented. Discourse representation for question-answer pairs (and other kinds of dialogue structure) is in progress, not yet implemented.

8.2.1.11 Events and Event-Type Anaphora

The implemented formalism uses a Davidsonian event-based semantics, and offers some treatment of event and event-type anaphora (again, these have a high frequency in the domain and are central to the task). A really adequate treatment of this kind of phenomenon depends on a workable theory of events and situations, which is still a desideratum.

8.2.2 Computationally relevant aspects

8.2.2.1 Interface to syntax

Semantic construction uses λ-DRT with generalized functional composition. This allows a strictly compositional construction of meaning representations, and at the same time satisfies discourse semantic requirements. The VM demonstrator architecture is strictly modular and largely sequential. There are two alternative syntax components, HPSG (IBM) and LKP (Siemens). Semantic construction communicates with syntax via abstract interface predicates (ADT approach). Thus it can access arbitrary syntactic information without being committed to a specific syntactic representation theory. (The interfaces to other components - e.g., transfer, generation - also use the ADT technique, so as to allow different views on the semantic structure.) The concept of the syntax-semantics interface also allows for non-sequential processing. The VM prototype will do linguistic analysis in an interleaved mode, allowing bi-directional information exchange between syntax and semantics.

8.2.3 Underspecification

Semantic construction provides representations which are underspecified with respect to scoping (Cooper Storage Technique), anaphora and ellipisis ("alfa" and "epsilon" expressions with uninstantiated binding feature), lexical ambiguity: Vague and ambiguous expressions are represented by an unspecific semantic predicate, and paralleled by a "concept feature", which can be used for successive contextual specification. Since speech input leads to drastic underspecification, due to the uncertainty of, or complete lack of, linguistic input information, the semantic formalism will be extended for "radical underspecification" cases. This is not yet implemented.

8.2.3.1 Contextual Reasoning

So far, only local disambiguation techniques are implemented. General research on local and non-local contextual resolution techniques is under work.

8.2.3.2 Inference and Evaluation

A considerable part of the formalism has a compositional denotational semantics (modulo implementation of β-conversion in terms of first-order unification where λ-calculus variables are modelled in terms of the variables of a logical programming language where all variables are universally quantified by (implicit) quantifiers taking wide scope over a clause). Parts of the representation ar uninterpreted, and serve as additional input for local transfer. The inference system used is BACK, a KL-ONE like terminological reasoning system. Communication between semantic construction and inference takes place through an interface established by "concept features" which parallel all atomic semantic information (predicates, discourse referents). Because of its restricted expressive power, the inference system cannot make full use of the information contained in the semantic representation. A full first-order (or higher-order) theorem prover is not yet available.

8.2.3.3 Lexical semantics

The VM Demonstrator has a domain-specific lexicon with 1500 word forms. In addition to type and structural semantic information, the semantic lexicon provides detailed sortal information (which is related to the domain model), thematic role and subcategorization information.

Chapter 9

Classification Dimensions

9.1 Introduction

In this chapter we give an overview of how semantic theories of natural language can be classified. The chapter is also meant as a short informal exploration of the different angles from where harmonization of formal approaches to natural language semantics can be initiated.

9.1.1 The structure of the chapter

In this introduction we will briefly sketch a somewhat over-simplified perspective of the development of semantic theories. This schema can be used to distinguish and classify different semantic theories on the basis of how they were constructed. Before we reach that point (in section 9.2 and 9.3), we will first give a short classification of the various interest groups who are involved. In fact, the classification dimensions of semantic theories emerge from the varying perspectives of these interest groups. In section 9.4 we present a list of key issues for discussion among semanticists. We try to relate these issues to the more historical perspective of 'semantics in action' of subsection 9.1.2. We also briefly indicate what different theories have to say about these topics. For practical reasons, we have used the DRT-perspective as an exemplar here.

In the final section (9.5) we give an indication of how the classification discussion relates to some central topics taken from the list of phenomena of the D5-

fragment. It is our hope that further exploration of these issues will lead to a unification of the approaches in phase II of the FraCaS-project.

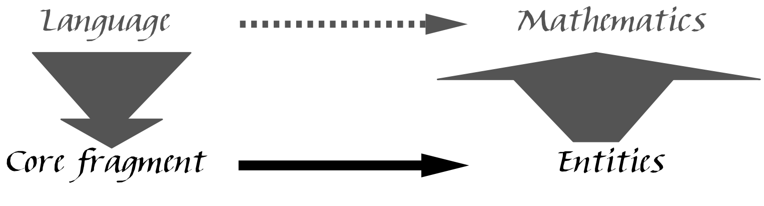
9.1.2 Semantic chronology

The science of formal semantics of natural language can be concisely described as the quest for appropriate 'meaning mappings' from natural language to mathematical structures.

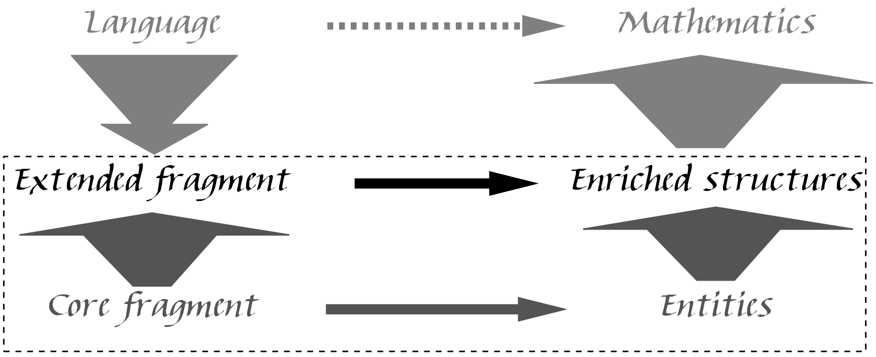
Language
Mathematics

The mathematical structures provide the formal toolkit which semanticists need to express themselves in an objective fashion and to check the consequences of their theories in a precise manner. Furthermore, the mathematical interpretation of natural language brings us one step further in the direction of computational feasibility. Formal understanding of natural language points the way to implementation of inference engines for natural language.

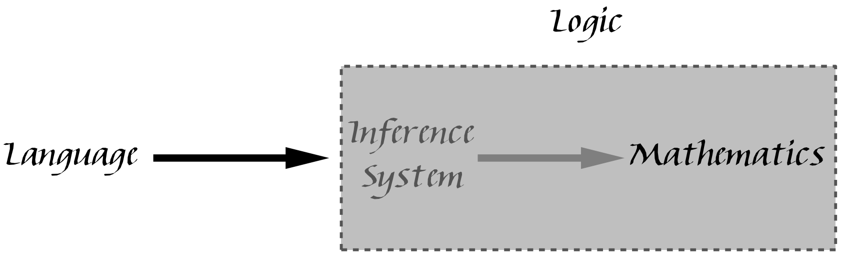
Of course, the semantic enterprise for natural language is extremely ambitious. Putting complete natural languages into a mathematical format is something that semantic theorists may dream of, but in practice they try to focus on a core fragment of the language which incorporates phenomena that they think are essential for its speakers to exchange information about a specific application domain. By analytic philosophic inspection they try to force this fragment into a part of mathematics. This restricted analysis yields the basic mathematic entities that they construct their theories on.



If this analysis has led to a satisfactory mathematically stable formalization, semanticists start to make efforts to extend the fragment with additional linguistic data. The success of the core theory depends on the possibility and convenience of extending the mapping which is imposed on the core fragment in order to give a formal account of natural language phenomena outside this core. In case of failure, theorists have to adapt the formalization in such a way that the analysis of the core fragment is still satisfactory and at the same time 'extendibility' is taken care of.



This picture suggests that formal semantics is not a trivial enterprise. Fortunately semanticists are not working all alone. Historically, neighbours are mathematical logicians and analytic philosophers, who have been studying the same field and have developed the science of formal logic in the process. Since the days of the foundational debate on valid mathematical reasoning this field has generated an impressive amount of knowledge on mathematical model theory and corresponding deduction systems for reasoning about such structures.



Very often formal semantic theories relate their model-theories to some known logical system. The task that is left then is to give appropriate translations from natural language to the logical language of choice. In many cases however, the mathematical structures that logicians use to model formal reasoning are ill suited for natural language semantics. Important phenomena of natural language do not fit easily in standard systems of formal logic. This sometimes leads to new combinations of logical systems. An outstanding example of this is Montague's formal semantics, which combines higher order logic with intensional logic. But it would be misleading to say that natural language semantics always has to borrow from mathematical logic. In fact, semantics has exerted a marked influence on logic too. Modern logic is focussing on alternative model-theories and one of its main motives for doing so is natural language semantics. Other motives for modern logic to widen its scope are provided by formal cognitive science and theoretical computer science.

In this paper we will take the pictures above as the starting point to give a classification of semantic theories. In fact, the divergence of semantic theories can be understood roughly by scanning them as the consequences of divergence on different key spots in the diagrams above. To start with, theories may be different in their core fragment input. Then, different analyses of this input yield different mappings and mathematical targets. These differences will be outlined in section 9.2.

Besides this formal differentiation, there are also more pragmatic inputs to formal semantics which make them diverge. First, there is the issue of the general convenience of the target logical language. That is, how (in)transparent are the translations from natural language to the target logical format? Secondly, how closely do theories approach the ultimate goal of actual implementation? In many semantic theories this issue is neglected for the simple reason that formalization is hard enough, and that the search of some computational part of the target model-theory is just a bridge too far. In the computational semantic enterprise nowadays this argument is no longer acceptable. In section 9.3 we will briefly discuss how current semantic theories can differ with respect to practicality and implementability.

9.1.3 Interest Groups

It is unrealistic to expect one neutral and general set of classification criteria, simply because different people have different aims and interests. From the outset, it is useful to distinguish three broad interest groups (by no means mutually exclusive):

1. Logicians:
   Clearly they classify semantic theories on the foundational issues of section 9.2 (e.g. paradox, intensionality, partiality, etc) They focus on areas of NL semantics that are logically or philosophically problematic. Most often they try to work on such problems on the basis of mathematical model-theory and they aim for meta-theoretical results.

2. Linguists:
   Linguists in the field are more interested in the expressive capacity of a semantic theory. Coverage is the most crucial criterion for semantic theories from the viewpoint of linguists who try to give a formal account for their empirical data. Of course, this criterion is only a very rough guideline for linguistic judgments of semantic theories. In practice, matters touching on the convenience of the formal encoding of data in such a theory also play a role. In other words: How much effort is needed to encode data in the format of the given theory? And: Does the representation still bear a reasonably close resemblance to the original NL input?

3. Implementors:
   This group takes a slightly different look at the practical issues of section 9.3. While linguists like great expressive power, implementors are faced with the computational price of it. In practice, implementors work with programming languages that do have a lower expressivity, and they may be forced to throw much of the expressivity overboard again in making the step from formal semantics to implementation code.

While these groups can and do overlap, what is important for one may be of marginal importance to another. The set of classification dimensions should be the union of all those relevant to the individual interest groups, not their intersection.

9.2 Foundational Issues

This first section relates to the first two diagrams in the introduction. There are three separate categories with respect to the semantic procedure in the second diagram. First, the focus on the core fragment which constitutes the backbone of a semantic theory. Secondly, the philosophical analysis of this fragment, and thirdly, the mathematical result of this analysis.

9.2.1 Motivations

As said in the introduction, semanticists and philosophers of natural language motivate their key theoretical choices on certain phenomena in natural language semantics which they think of as having key importance. For example, Montague semantics took *intensionality* as the most important issue of its formal setting [Montague, 1973]. In this respect Montague semantics evolved from the same concerns as situation semantics [Barwise and Perry, 1983] and property theory [Bealer, 1982]. These latter theories deviate from the Montagovian line in the way they treat so-called hyper-intensional phenomena (epistemic attitudes and perception verbs). According to the situation theorists such phenomena call for interpretation over partial possible worlds (situations) rather than the completely specified worlds which are used in Montague semantics [Barwise, 1981]. In property theory propositions inside the scope of subjective attitudes are taken to denote completely opaque objects. 'Internal' inference has then to be encoded by means of explicit axioms [Turner, 1992]. Very closely related to this central role of intensionality is the appearance of paradoxes in NL. In situation theory this has led to the adoption of non-well-founded sets for the representation of attitudes [Aczel, 1988] [Barwise and Etchemendy, 1987].

DRT, File change semantics and other dynamic semantic theories share a common motivation in a similar way, namely a concern with indefinite introduction of individuals in discourse and subsequent anaphoric reference to those individuals [Kamp, 1981b] [Heim, 1982] [Groenendijk and Stokhof, 1991]. What the theories have in common is that this dynamic aspect is taken as a cue for deviating from the safe (and static) path of standard logic. All of these theories more or less try to combine static semantics with an analysis of dynamic aspects of language use, akin to the Hoare-Floyd style semantics of imperative programming languages. Indeed, there is a close connection between the use of variables in imperative programming and the use of 'discourse markers' to provide hooks for subsequent anaphoric reference.

The key motivation for monotonic semantics, finally, is the wish to deal in a satisfactory way with ambiguity and vagueness [Alshawi, 1992].

9.2.2 Philosophical issues

Of course, the motivational part as outlined above is strongly connected with the basic philosophical line taken in a semantic theory. For example, Montague semantics takes the line that natural language semantics should use Tarski style models (Tarski [Tarski, 1956]) and should adopt Kripke's theory of naming and necessity [Kripke, 1972] as a way of dealing with the problem of 'trans-world identity'.

Situation semantics can be seen as an initiative towards a more cognition oriented approach to formal semantics. This has lead to a drastic deviation from the standard logical style of Montague semantics. More recently, various semanticists have shown that situation semantics can also be reformulated in a more standard logical fashion, and will then more closely resemble Montague semantics (see Muskens [Muskens, 1989], Escriba [Escriba, 1992]).

The core philosophy of DRT and its dynamic neighbours can be summarized with the slogan 'meaning as change'. The meaning of a sentence is no longer identified as the set of models which satisfy its content, but should be taken as the way an interpreter (should) change(s) his mind when this sentence is successfully processed. This deviation leads to a set-up of the semantic definitions in which 'truth' becomes a derived notion. The logical constants receive a dynamic redefinition, and a lot of effort is spent on investigating basic semantic notions such as validity and entailment in the new light of the dynamic viewpoint.

Property theory is based on a 'hyper-fine grained' philosophy of intensionality. It treats propositions in the scope of intensional verbs as logical objects, and consequently, as completely opaque with respect to 'external' inference. This approach has a number of advantages for the truth-conditional and epistemological puzzles of analytic philosophy (see [Bealer, 1982]).

9.2.3 Mathematical issues

The philosophical divergence of semantic theories also generates a technical mathematical separation. Montagovian semantics is traditional logic combined with modal logic. Initially, situation semanticists tried to combine all this with partial model-theory. In a later stage, non-well founded sets have also been used to give a satisfactory analysis of semantic paradoxes that arise from self-referential sentences. Dynamic semantic theories have generated a good deal of interest in relational logic for formal semantics. Sentences are taken as modifications of the interpreter's state of mind. Technically speaking, a sentence is taken to be an input-output relation which describes this modification. The hyper-fine grained analysis of property theory has led to the position that NL semantics can be done in first-order logic over a domain which includes objects representing the propositional arguments of attitudes. Montagovian types then become first order sorts (subsets of the domain). This causes a first-order flattening of the Montagovian viewpoint. Monotonic semantics has challenged another part of standard logical analysis of natural language. Instead of taking sentences to be interpreted over incomplete structures as in situation semantics, monotonic semantics takes this proper interpretation to be partial. Underspecification in natural language sentences is no longer taken as a cue to generate different logical interpretations. Rather, interpretation may now stop at a point where the output is still underspecified. The effect is that the mapping from a sentence to the intended model-theory is partial; it can be completed by further specification provided in later sentences.

9.3 Practical Issues

This section discusses two practical dimensions of semantic theories. First, what does a semantic theory have to offer to the 'programming' computational linguist? Secondly, how convenient is the semantic theory for the empirical linguist when he is looking for formal tools to give an account of his findings?

9.3.1 Computational Issues

Computational Issues are an important theme for comparison and harmonization in the FraCaS-project. There are two levels where semantic theories can be connected to questions of computation. First, their relation to computation practice, and second, their formal closeness to programming languages. An exponent of the first computational link is monotonic semantics. This theory evolved from the practice of computational linguistics. DRT and other dynamic semantic theories are examples of the second relation. In fact, their semantics of 'changing assignments' to deal with the dynamic role of discourse referents originates from the Hoare-Floyd style model-theory for imperative programming languages.

9.3.2 Empirical Issues

Empirical classification of semantic theories relies on the experiences of linguists working on real language data. How easy or difficult is it to find appropriate accommodation of NL semantic phenomena in formal semantic frameworks? As said in the introduction, semantic theories most often originate from the philosophical and logical analysis of certain key phenomena. This may lead to inconvenience with respect to giving a satisfactory analysis within the given framework for phenomena which are outside its initial scope. A possible way for FraCaS to give a classification of semantic theories is to compile a check list from the D5 fragment and inquire on possible solutions which are given in different frameworks.[1] Of course, such an approach would not generate strict judgments about theories. Most often, theories themselves do not explain phenomena. Such a test would only give an up-to-date perspective on the practical use of theories among linguists, and should therefore be valued as the outcome of a momentary poll.

9.4 Classification Questionnaire

We will now scrutinize two of the frameworks (Property theory and DRT) treated in FraCaS with a questionnaire touching on a number of classification dimensions which are suggested by the discussion above. Note that some dimensions fall under more than one of the headings distinguished above. The two sets of answers are included below as an illustration of how semantic theorists look at the classification dimensions. We hope to say more about this whole issue in phase II of the project.

Partiality To what extent, and in what ways, does the theory deal with the inevitable partiality of information one has about the world? Issues here can range from the foundational (partial situations as objects of attitudes, information extension in the treatment of epistemic modalities), through the linguistic (partial mutual or common knowledge in dialogue interpre-

[1] For further details on this we refer to deliverables D2, D8 and D9.

tation), down to quite low-level implementation concerns (closed world assumption).

This dimension falls under the headings 'mathematics' and 'computational'.

- Partiality, in Property Theory with a classical foundation, can be treated via incompleteness: a particular information state will fail to allow the truth conditions (or even the propositionhood) of the representations of some sentences to be inferred.
- There is an obvious sense in which information is almost always partial: Only in the rarest cases (and these will necessarily concern abstract, not concrete things) will what one knows, or what one is told, exhaust all that is true about the topic in question. Classical logic has always allowed for this, and allowed for it in the obvious way: Information is assumed to take the form of a (logical) theory, which is "partial" iff it is not complete; and this partiality is manifest in that the theory has at least two non-equivalent models (the most common situation from the point of available information about real individuals is of course that where the corresponding theory has lots of non-equivalent models).

  DRT has allowed for partial information in essentially this same sense, in that the structures it proposes for the identification of information - DRSs - are like logical theories in this respect. In particular, a DRS that acts as a representation of context, or alternatively as the knowledge available to the interpreter of an utterance, will as a rule be partial in that it is compatible with an indefinite variety of different models or possible worlds; and the same goes for the result of interpreting this utterance - the new information state represented by the "output DRS", which results from incorporating the information contributed by the interpreted utterance into the input DRS.

  A second sense in which DRT allows for partiality is one pertaining to "updating" of a context by a new sentence or utterance (see also comments on next item). From the start, DRT allowed for the possibility that certain contexts could not be updated by certain sentences (e.g. because the sentence contains an anaphoric pronoun for which neither it itself nor context provide a suitable antecedent.) More recently this kind of partiality has become more important and more clearly visible through the DRT-based work on presupposition ([Zeevat, 1992], [Sandt, 1992], [Geurts, 1994], [Sæbø, 1993], [Roßdeutscher, 1994]). Presupposition failure may trigger the back-up strategy of accommodation. But in those cases where accommodation is blocked too the result is undefinedness in the present sense.

Finally, a third element of partiality has been introduced into DRT through the work of Reyle on Underspecified DRSs [Reyle, 1993a]. UDRSs are structures which may leave certain kinds of information (e.g. about the scope relations between quantifiers) underspecified, whereas the DRSs of "standard DRT" are required to make this explicit. Thus to each UDRS corresponds a set of DRSs, the members of which resolve the underspecifications of the UDRS in one of the various possible ways; the degenerate case is that where the set is a singleton and the UDRS in essence a DRS. Underspecification can either be used to leave ambiguities unresolved which the interpreter (human or machine) would like to be able to resolve but can't because of insufficient evidence; or it can be adopted as a way to meet the demands of those who reject the logical formalisms propounded by DRT and other forms of formal semantics as having expressive power going beyond the semantic discriminations made by most or all human speakers.

Context Change Sentences can update information (contexts). What does the theory have to say about the context update potential of sentences. What are contexts? Is there a distinction between contexts and models (e.g. linguistic expressions can be part of context, but not of a model of the world).

- Property Theory with dependent types models context as a structured term that is passed on as an argument to the representation of subsequent parts of the discourse [Fox, 1994].

  The context change potential of sentences and noun phrases etc. is modelled by allowing their representations to amend the structured term representing the context for subsequent discourse.

  Contexts are formally distinct from models.

- Capturing the update potential of sentences has been one of the central concerns of DRT from its inception. In this regard it is on a par with File Change Semantics [Heim, 1982] as well as with the more recent forms of Dynamic Semantics, such as Dynamic Montague Grammar [Groenendijk and Stokhof, 1990]. Although early DRT has failed to go on record on this in the same explicit way as File Change Semantics, which emphatically identified the meaning of sentences with their "File Change Potentials", the same idea is quite clearly visible within DRT, as one of the central semantic operations defined there is that of transforming a given context DRS K into a new context K'

through incorporation of a new sentence s; this is nothing more or less than associating with each sentence s (of the relevant language or fragment) a function U (in general U is both partial and multi-valued) which transforms DRSs into DRSs.

With this function U one can then, if one so wishes, associate a further (partial, multi-valued) function U' from sets of possible worlds to sets of possible worlds such that if X and Y are such sets, then $U'(s)(X) = Y$ iff there are DRSs K and K' such that K represents X, K' represents Y and $U(s)(K) = K'$. However, as the DRT literature has endeavoured to make clear form the start, this further step is problematic, since it is quite common for K_1 and K_2 to represent the same world set X while $U(s)(K_1)$ equals Y but $U(s)(K_1)$ does not. Thus from the perspective of DRT world sets are too coarse-grained as the "information states" that sentences can profitably be regarded as updating. Many model-theoretic constructs have been and are being invoked to resolve this problem, starting with Heim's notion of an information state as a set of pairs consisting of a world and a partial assignment. The most recent (and most sophisticated) efforts in this direction ([Elworthy, 1993], [Krifka, 1994]) seem to raise a legitimate question whether this will not lead in the end to a full reconstruction of DRS structure at the model-theoretic level.

Implicit in these comments are answers to the remaining questions raised under the heading "Context Change": The contexts of DRT are not models, they are either DRSs or, alternatively, contexts are intension-like model-theoretic objects of any one of the kinds alluded to in the last paragraph.

- Falls under the heading 'philosophy'.

Context Dependence In addition to changing context, the interpretation chosen for a sentence typically depends on its context of utterance. What does the semantic theory have to say about context dependent aspects of interpretation, and how closely does this tie in with the notions of context change above?

- As described above, Property Theory with dependent types treats contexts as arguments to the representations of components of the discourse. Terms in the context can be selected as the antecedents of anaphoric expressions. Resolution of anaphora can be blocked in some cases (e.g. singular anaphoric reference to a universally quantified antecedent) by the nature of the context provided by the preceding discourse.

- One of the original tenets of DRS was that natural languages contain certain elements (the first examples were the past tenses of French) whose meanings (= semantic contributions) can only be properly understood in terms of the instructions they carry as to how semantic material in the sentences containing them is to be connected with the contexts in which the sentence is being used; and, moreover, all other semantically significant elements of the language could also be analyzed along these lines (even if for many of them there appeared to be no compelling reason to proceed in this fashion). This led to the conception of an explication of meaning through a combination of the DRS construction algorithm together with a model theory for the resulting DRSs. Context change (in the sense of updating) and context dependence (the proposition expressed by a sentence cannot be determined in isolation of the context, including the verbal context, in which it is used) arose as two distinct, though intimately connected and natural side-effects of this way of proceeding.

  In recent versions of DRT, which abandon the top down construction algorithm in favour of a two step procedure, which first constructs a "local" DRS just for the sentence that is being processed and only then, often after certain intermediate operations needed for the verification and/or accommodation of its presuppositions, integrates this DRS into the context (cf. [Sandt, 1992], [Kamp and Roßdeutscher, 1994a], [Kamp and Roßdeutscher, 1994b]) the separate issues of context dependence and context change are more easily visible than they are within the original DRT architecture (as in [Kamp, 1981b], [Kamp and Reyle, 1993]).

  An especially interesting aspect of context dependence is the dependence on "local" contexts - as for instance the interpretation of the consequent of a conditional may depend on the "local context" created by its antecedent. This kind of dependence has been the motivation behind DRT's treatment of, among other things, conditionals and quantification, and, more recently, behind the work on the projection problem in presupposition theory ([Sandt, 1992], [Geurts, 1994]). In particular, the notion that the restrictor of a quantificational structure is, just like the antecedent of a conditional, a (partial) description of a situation, event or state of affairs which is expanded by the description provided by the nuclear scope (as is the description provided by the antecedent expanded by that provided by the consequent). This analysis of natural language quantification entails that natural language quantifiers are always conservative (i.e. the quantifier, as a binary relation between satisfaction sets A and B, has the property that it holds between A and B iff it holds between A and $A \cap B$), a claim which seems to be well supported by the facts.

Domain Dependence The interpretation chosen depends on the domain of application. What does the semantic theory have to say about the way in which the domain constrains interpretation?

- Property Theory itself is domain independent (so long as the domain does not require the full power of higher order representations).

  One existing application that uses Property-theoretic semantics for a natural language front end to databases (Squirrel) keeps the front end domain independent [De Roeck *et al.*, 1991e]. Only when the representation is translated into a database query is knowledge of the domain used to disambiguate the representation of the natural language query, and eliminate unacceptable readings.

- This is a topic that DRT has so far had nothing to say about. The position has been that, as a theory of natural language semantics it should not be in the business of making claims about how interpretation is affected by information about particular application domains, or the presumably extra-linguistic knowledge which language users may have about it.

  At the same time DRT's position is that the architecture of a semantic theory should show where and how the interpretation process can make use of extra-linguistic knowledge if and when it needs to. The architecture of recent versions of DRT make provisions for this insofar as they explicitly allow for inferences in the course of DRS construction; these inferences may use parts of the context DRS and the (provisional) DRS as premisses, and may also draw upon extra-linguistic knowledge to supply additional assumptions. (Such inferences are needed, for instance, for the resolution of lexical and syntactic ambiguities, for the resolution of anaphora and for the verification and accommodation of presuppositions. (For illustrations of such inferences see in particular the work of Asher, Lascarides and others [Asher, 1993], [Lascarides and Asher, 1993], [Lascarides and Oberlander, 1993] on the identification of rhetorical relations in the construction of S(egmented) DRSs)

Ambiguity and Vagueness How does the theory handle ambiguity (underspecification, generate and test?). Is ambiguity distinguished from vagueness? Does ambiguity and/or vagueness need to be resolved before the theory can specify inferential relations between sentences.

- Property Theory can treat ambiguity and vagueness in the semantic representation by having weak axioms that allow different proofs of the truth conditions for the representation [Fox, 1993]. Some cases of ambiguous quantifier scoping can be dealt with in this way, for example, as can instances of ambiguous properties and relations.

  In principle, some inference can take place using the ambiguous representations.

- Some of the issues here have been considered tangential to DRT. But as the fact that DRT has not been concerned with them is a reflection of how the theory has perceived its own de-limitations, a comment seems nevertheless in order.[2]

  DRT is based on the assumption that there is an important and more or less clear distinction between ambiguity and vagueness. Its business is with the former, but not with the latter.

  The distinction between ambiguity and vagueness that is assumed here is roughly this: ambiguity is incompleteness of interpretation; vagueness is truth-conditional underspecification of (full) interpretations. Let me explain.

  First ambiguity. An expression is ambiguous iff it allows for more than one distinct interpretation. This "definition" presupposes a formally precise notion of interpretation for NL expressions (words, semantically significant morphemes, phrases, sentences, discourses, texts). So the notion of ambiguity has an inevitable theoretical component: it acquires a definite meaning only within the context of particular formal semantic theories for particular languages.

  A consequence of this definition is that ambiguity can manifest itself in two different ways. First, an expression may be ambiguous on its own, but lose its ambiguity within the context in which it is used. This is an ubiquitous feature of natural language and one of the reasons why they work as well as they do (in fact: why they work at all). A very substantial part of the vocabulary of natural languages is ambiguous - presumably there are important cognitive reasons for this, having to do with efficiency of storing, retrieval, acquisition and/or the lexicalization of new concepts. Yet, most ambiguities disappear as the meanings of words are combined into the meanings of the phrases and sentences containing them, or when those get integrated into coherent interpretations of the texts and discourses that are made up of them. It is one of the most significant hum-drum facts about human

[2] At the same time, however, these comments reflect my personal opinion on certain issues not visibly related to DRT and should be taken in this spirit. (H.K.)

natural language understanding that people fail to notice almost all the *local* ambiguities of the constituents of the utterances they interpret, because they get filtered out at the level of the interpretation of the utterance as a whole. And it is one the most important problems for computational semantics to analyze how disambiguation is achieved in the course of the composition of meaning. It is also one of the most difficult and, at the present time, one of the most underexplored problems; its solution will require, among other things, a much tighter and much more precise integration of (what are now called) semantics and pragmatics than is currently the case.

Secondly, there are of course also the cases where ambiguities survive, at the level of the complete sentence or even of that of the cohesive discourse or text. In the linguistic literature there has been a tendency to concentrate on ambiguity of this hardy variety. In the light of what has just been said, however, the ambiguities that remain when all has been said (by the speaker) and done (by the interpreter) are the anomalies; they are symptomatic of the deep and pervasive underlying problem; but it is only in a comparatively minor sense that they pose a problem themselves.

Still, they do pose a problem. Sometimes the evidence at hand is not sufficient to construct an unambiguous interpretation of the kind the semantic theory requires. (This may be because our theory or its implementation aren't up to scratch; but as we said before, certain sentences and texts just are ambiguous, so that no improvement in theory or implementation would relieve the problem.) In these cases we need to be content with a "provisional" representation. This, as we said under the heading partiality, is one of the things underspecified DRSs [Reyle, 1993a] are for.

But the real problem, we said, is not that of representing ambiguity but of resolving it. This is a problem which DRT has begun to address only in the last few years ([Kamp, 1992], [Kamp and Roßdeutscher, 1994a], [Kamp and Roßdeutscher, 1994b]), showing for instance, how certain ambiguities can be resolved through inferences triggered by the need to identify anaphoric antecedents or to justify presuppositions.

So much for ambiguity.

By vagueness we understand a property of semantic primitives which manifests itself as indeterminacy of extension. The current position of DRT is that the semantic primitives which enter into the meaning of a sentence or discourse are represented as components of its DRS. Thus, the vagueness of a primitive is a matter of the relationship between the DRSs in which it occurs and the models in which they are evaluated for truth and falsity, in other words of the model theory for DRSs. While this is of course no less a legitimate issue for DRT to worry about than it is for any other semantic theory, it is a matter that has not been taken up in there, as most of the issues which it raises appear to be disconnected from the features that distinguish DRT from its semantic rivals.[3]

There is one aspect to vagueness, however, which might well profit from an analysis in a DR-theoretical setting. As argued in [Kamp, 1981a], and again along somewhat different lines in [Pinkal, 1985] and [Pinkal, 1991b], one feature of vague expressions, which among other things plays a crucial part in the Sorites paradox, is their "context shaping potential": if in a given context c, the object a is an indeterminate case of the vague predicate P, so that in this context $P(a)$ counts neither as definitely true nor as definitely false, it may nevertheless be possible to assert $P(a)$. By doing this the speaker expresses his presumption that a does belong to the extension of P and provided his audience does not signal disagreement, the context thereby counts as amended to one relative to which a belongs to the extension. It would seem natural to formalize this intuition in a DR-theoretical setting, in which context DRSs also encode such local commitments to the extensions of vague concepts.

Compositionality Is the semantics built up compositionally off syntax? If so, how are contextual and other forms of ambiguity handled. If not, what is the theoretical and/or computational price-tag of abandoning compositionality?

- Property-theoretic representations can be built compositionally off the syntax. Ambiguity can be dealt with by "generate and test" or by having weak axioms for the truth conditions.
- One of the most consistent criticisms of DRT has been that it is "not compositional". Although the ensuing discussions (See e.g. the DYANA reports) have clarified the matter somewhat, differences of opinion have not been fully resolved; and what is more, it would seem that certain confusions, pertaining to what compositionality is and to what qualifies a theory as having or lacking it, keep raising their heads. The problem is that compositionality has meant a number of different

[3] Some of my own views on vagueness have been published elsewhere, where they are discussed in frameworks other than that of DRT ([Kamp, 1975], [Kamp, 1981a], [Kamp and Partee, 1995]).

things to different people and that its formal definitions have been neither self-explanatory, nor, for that matter, theory-independent.

One familiar notion of compositionality (it is sometimes called *strict compositionality*; perhaps a better name for it would be *rule-by-rule compositionality*) is one that is manifest in Montague Grammar: with each rule of syntactic composition is associated a rule of semantic composition, which computes the semantic value of the syntactic compound out of the semantic values of its syntactic components. A particular version of this notion insists in addition that the semantic rules always consist of one or more function applications (as required in Basic Categorial Grammar; call this *function compositionality*). It has been argued that DRT, in its original top down formulation, fails to be compositional in this sense. There is of course a fairly straightforward, but uninteresting sense in which this is true, for a top-down algorithm like DRT's clearly does not provide a compositional account directly. However, the criticism has a more probing construal than this. In principle it might have been possible that with the construction rules, which decompose syntactic compounds into semantic representations of their syntactic constituents, one could have associated, more or less straightforwardly, rules which combine the semantic values of those out-put representations into the semantic value for the constituent that the construction rule reduces. But evidently there is no *straightforward* way of doing this. For the semantic rules associated with each of the syntactic rules transform DRSs into other DRSs; but in general these DRSs are so-called "intermediate" DRSs, while the model theory of DRT associates semantic values only with the completed DRSs. This means that in general there are no semantic values associated with input and output of the "semantic" rules; a fortiori there is no way to reinterpret the rules themselves as functions that map semantic values to semantic values or to associate such functions with them.

Indeed, it was part of the original conception behind DRT that compositionality of the kind that the theory has been criticized for not proffering, is not necessarily the Summum Bonum it is sometimes made out to be. The ultimate reasons for wanting something like the compositionality which Montague required (and surely those reasons are compelling) derive from the insight that only a finitary statement of the syntax and semantics of a language could explain how such a language could be learned and known. But there are many different forms that such a *finitary* description could take. Descriptions that take the form of a Montague Grammar are one kind, but there are many others and DRT (whether in its original formulation or some later version of it) is among the alternative options.

The first publications on DRT, however, seemed to carry not only the message that its architecture was an acceptable alternative to compositionality, but that the linguistic facts they focussed on (in particular: trans-sentential pronoun bindings by indefinite antecedents) force such an alternative architecture upon us. In part the criticism was a challenge of this apparent claim. This challenge was certainly well motivated. For as the relevant publications ([Barwise, 1987], [Rooth, 1987] and [Groenendijk and Stokhof, 1990]) showed, one can give a coherent account of these phenomena in a compositional setting, if only one is prepared to complicate the concept of the semantic value of an expression. It is important here to stress that the extra complexities which these compositional treatments of trans-sentential anaphora need, are of two distinct kinds (it is the interaction between them which produces the desired treatment.) First, semantic values have to be made relational, consisting of an input and an output; this, many people have said, is the essence of a dynamic theory of meaning; indeed it is very closely connected with Heim's idea of meaning as File Change Potential, and arguably already contained in that. In addition, however, it is also necessary to complicate the input and output values themselves -for instance, where the expression is a sentence, they can no longer be just sets of possible worlds, but must be sets of pairs $\langle w, a\rangle$ where w is a world and a a partial assignment in w. It is only when values are complicated at least this much (i.e. when they are enriched with assignments or embedding functions) that the relational concept of meaning is going to do any work. But if this is the only change we make to the input and output values, then we can deal with the original "donkey" problem, but there will still be many other problems of discourse semantics that cannot be treated and which will require further complications. (We already referred to the work of Elworthy [Elworthy, 1993] and Krifka [Krifka, 1994] as showing how much more structure is needed, if one is to deal with plural pronoun anaphora along these lines.)

The conclusion is that while the critics were right to challenge the suggestion that the donkey problem could not be treated in compositional fashion, their proof that it can be done is nevertheless an indication that one has to abandon some of the assumptions that semanticists had been making until then about the semantic values of expressions of given categorial types.

More clearly situated within DRT are several formulations of DRS-construction, which build DRS-like structures for sentence constituents and combine those for the daughters of a given constituent into a semantic representation for the mother. Such algorithms look very much like the compositional definitions of semantic values that are the trade mark of classical Montague Grammar. Examples are the bottom-up algorithms of [Asher, 1993] and [Zeevat, 1989] and, more recently, the λ-DRT of [Millies and Pinkal, 1993] and [Bos *et al.*, 1994b]. It is not always possible to reinterpret these algorithms in the sense of a compositional model-theoretic semantics. For instance, if DRSs are model-theoretically interpreted in a natural way, how are we going to distinguish between the value of the DRS for a proper name like *Carter* and that for the DRS of an indefinite NP such as *an American envoy*, if both these DRSs consist of a discourse referent together with one or two constraining conditions; as in (9.1) (a) and (9.1) (b):

$$(9.1) \qquad (a) \quad \begin{array}{|c|}\hline x \\ \hline \mathit{carter}(x) \\ \hline \end{array} \qquad (b) \quad \begin{array}{|c|}\hline y \\ \hline \mathit{american}(y) \\ \mathit{envoy}(y) \\ \hline \end{array}$$

Something in the value determined by (9.1) (a) should make clear that the discourse referent x is to end up in the main universe of the DRS which will eventually result (which entails that this discourse referent should be given widest possible scope). The value associated with (9.1) (b) should not have this property, certainly not as a matter of generality. But it is hard to see how naturally chosen values could encode this distinction. For sure, it is always possible to distinguish between types of discourse referents by introducing enough "function layers". For instance, suppose that the DRS for a town between Hamburg and Bremen is as in (9.2) with two "proper name discourse referents" x and y to represent Hamburg and Bremen and an "indefinite NP discourse referent" z to represent a town between them:

$$(9.2) \qquad \begin{array}{|c|}\hline x\ y\ z \\ \hline \mathit{hamburg}(x) \\ \mathit{bremen}(y) \\ \mathit{town}(z) \\ \mathit{between}(x,y,z) \\ \hline \end{array}$$

we might then, to capture the distinction between these two types of discourse referents, associate (9.2) with a function F from assignments of individuals to x and y such that for each such assignment a such that $a(x)$ is Hamburg and $a(y)$ is Bremen, $F(a)$ is a function from assignments b extending a by a value for z such that $b(z)$ is a town between $a(x)$ and $a(y)$. In this manner we can of course build in distinctions between any number of discourse referent types. Once the difference between proper name discourse referents and indefinite NP discourse referents has been marked, in this artificial way, by this kind of "blockwise currying of variables", we can exploit the thus encoded differences when stating the composition rules which combine these values with others. It can be done, but it is not very appealing.

As regards price-tags: From what we have said so far, it rather seems that within DRT there is a price attached to those approaches which try to stick coute que coute to compositionality of the strict model-theoretic variety. It is also true, however, that some work in DRT has proceeded with a rather blythe disregard for the formal constraints that go hand-in-hand with a proper concern for compositionality. Thus there has for a number of years been a need for charting in some principled way, the different operations that DR-theoretical investigations have seen fit to postulate for the description of certain types of linguistic phenomena. The systematic criticisms of these forms of DRT and, more importantly, the more principled reformulations of substantial parts of the theory have already provided much insight into the different formal constructors implicit in the DR-theoretical modes of linguistic description and explanation. This is especially important from a computational perspective, as it furthers declarative formulations of the theory, which provide a clearer picture of what an implementation is to achieve, without unnecessarily deciding certain matters of procedural architecture in advance.

Representationalism Does the theory posit an intermediate level of representation? What motivates its introduction (e.g. representational treatment of intensionality, contextual ambiguity and compositionality)? Does the precise syntax of the intermediate representation play an unacceptably large role?

- Under the conventional interpretation, Property Theory has an "intermediate" level of representation between the natural language expressions and the model of the semantic theory. This level of representation plays a central role in the theory, and is motivated by issues in intensionality. It also allows a treatment of ambiguity and presuppositions via weak axioms for the truth conditions [Fox, 1993] and weak axioms for propositionhood [Fox, 1994], respectively. This level of representation is precisely that required in an effective implementation.

- Much of what might be said about representationalism within DRT is closely connected with the issue of compositionality and most of the important points have already made under that heading.

  Originally DRT ventured to claim that the account it offered of the donkey phenomenon was possible only in the representational setting it had adopted - in which the manipulation of representational structures (the reducible and irreducible DRSs) only some of which (the irreducible ones) were model-theoretically interpretable, played the central role. Like the (closely connected) claim that one had to abandon compositionality, the unqualified version of this claim has been shown untenable. We already noted, however, that, as the repertoire of linguistic phenomena grows for which DRT has reasonable treatments to offer (especially those at the supra-sentential level), much of the ensuing complexity simply has to be shifted into the semantic values that are needed in order to eliminate representationalism from those treatments.

  How much this issue - which tends to exercise formal semanticists to an often surprising degree - matters to computational linguistics is a moot point in any case, as all computational linguistics is of necessity about representations of one kind or another; in particular, computational semantics is about semantic representations. True, there is one consideration which favours presentations which are non-representational, or which can easily be converted into non-representational versions: such presentations offer advantages form the perspective of portability. But as soon as we pass from such a general presentation to actual implementation, representations are the alpha and the omega, for they are all the computer can see (unless it is a robot; but that is a different story).

  In fact, representations are all-important within natural language semantics not only from this computational perspective but also for another reason, which has to do with the character of natural language meaning and interpretation and, more specifically, with the close connections between meaning and inference. The interpretation of dialogue and texts continuously involves the logical operations of monotonic and non-monotonic deduction. So - unless logical theory, from Aristotle till the present day, has not been consistently on the wrong track in its conviction that deductions are formal operations, which proceed by principles which concern the symbolic properties of the premisses and conclusions they connect, while abstracting away from all specific content - the information structures that are constructed in the course of discourse interpretation must be of the right logical form to serve as inputs to the inferences that must be drawn for the interpretation to go ahead. If there has been a lack of appreciation for this point, this has presumably been because so little attention has been played to the inferential aspects of interpretation and meaning. Undoubtedly the study of these aspects is one of the most urgent tasks for natural language semantics today. In fact, it is not only of prime importance for computational semantics, it also is an indispensable input to our understanding of human linguistic cognition.

- Comment: All theories have a level of representation. The key question here is whether the expressions which make up this level are fully interpreted.

Paradox and Intensionality How are phenomena like paradoxes of truth and predication and intensionality dealt with (connections to representationalism)?

- Intensionality is treated in Property Theory by way of its representationalism. At the weakest, propositions are equal only if their representations are syntactically identical (variables may be renamed). Stronger equivalences between propositions in opaque contexts can be enforced by additional axioms.

  The weak typing of property theory allows expressions with the form of logical paradoxes to be represented. However, propositionhood is defined axiomatically, rather than syntactically (that is, the theory only implements *weak* representationalism). The axioms do not allow the logical paradoxes to be shown to constitute propositions, and so inconsistency is avoided.

- This heading seems a little underspecified. There are at least three issues, all equally important in the history of formal semantics and logic, which appear to fit this conjunction perfectly:

  (i) the paradoxes of material implication and the development of the theory of strict implication and of modal logic.

  (ii) the paradoxes of logical omniscience and the development of hyperintensionality and other accounts of “fine-grained” intensionality.

  (iii) the paradoxes of self-reference and the intensional theories of propositions, such as Frege structures or Property Theory.

DRT has had commerce with all three of these issues, but in quite different degrees.

(i) First, DRT's early concerns with conditionals and quantification arguably clarified the status of natural language conditionals insofar as it made more clearly visible how common it is for such conditionals to involve quantification over variables of different types - individuals, events, times, states, situations. The moral (which can be and has been arrived at by many other routes as well) is that the strict opposition between material and modalized conditionals which is distinctive of earlier discussions of conditionals, is misleading. Rather than distinguishing two main types of conditionals the present perspective sees them as covering a multi-dimensional continuum of meanings, in which the "material-strict" opposition is only one among many.

(ii) The problem of omniscience has been one of the central motives behind DRT's theory of propositional attitudes and of the parts of language that are used to talk about them. The wish to deal with the problem of omniscience added extra impetus to a move which seemed a natural one within a DR-theoretical setting anyway, viz. to use and further explore DRSs as cognitively and linguistically relevant ways of classifying attitudinal states.

(iii) It is only a small step from this project to making DRSs themselves into the objects of the attitudes in a formal theory of attitude reports. This can easily lead to a theory which suffers from all the familiar pitfalls of self-reference, which had come into full view through the results of [Montague, 1963] and against which Thomason [Thomason, 1980] launched a crusade (much justified!) in the early seventies. It was argued by [Asher and H.Kamp, 1989] - a paper which neither uses nore mentions DRT, but which was motivated by the self-reference problem that arises when attitudinal predicates are construed as predicates of DRSs - that trying to escape the paradoxes by treating attitudinal concepts not as predicates of syntactic objects but as propositional operators, in the spirit of classical modal logic, is not a tenable solution. Instead the paper argued for the view that the logics governing various attitudinal predicates is extremely weak (and showed for a few sample cases, how weak).

This approach sticks to the classical conception that every well-formed formula of a formalism qualifies as expressing a proposition. In this it differs from a family of "constructive" approaches, which distinguish between well-formed formulas in general and those expressing genuine propositions, and which characterize this predicate (i.e. that which is true of a well-formed formula if it expresses a proposition) by prooftheoretic means. Notable among these approaches are in particular Aczel's theory of Frege structures [Aczel, 1980], the constructive type theory, e.g. along the lines developed by Martin-Löf [Martin-Löf, 1984], and the Property Theory of Turner [Turner, 1988], [Turner, 1992] and others.

Reversibility Can the semantic representation be used for generation? This is connected to the issue of declarativity. Is the semantic theory fully or partly declarative? Does it make appeal to separate processing modules (e.g. to fill in pronoun referents).

Processing Architecture Related to declarativity: does the semantic theory commit you to a particular processing architecture? Is this architecture usable in a variety of tasks, e.g. analysis *and* generation? (connection to reversibility).

- Property Theory says nothing about processing architecture. There may be a need for some canonical forms in order to simplify the process of generation of natural language from semantic representations.
- Much that is relevant to the question of processing architecture has already been said. What still deserves pointing out relates directly to the question of inference, which is why these two headings have been combined into one.

  We have already pointed out how important it is for a theory of natural language interpretation that it posit semantic representations (or "logical forms") which are capable of sustaining the inferences which have to be made in the course of interpretation (as well as those that are available after interpretation is completed). How such inferences are actually drawn is still too poorly understood. For certain DRS languages there exist complete or partial theorem provers that are attuned to the particular structure of DRSs (i.e. which explicitly exploit features that distinguish DRS languages from other formalisms that implement the fundamental concepts of predicate logic). But how closely these theorem provers correspond to the inference modules that are actually employed by the human interpreter is still anybody's guess. There are also emerging some insights into special inference modules such as those needed in many types of presupposition verification and accommodation, those needed to reason about tense and aspect or those operative in the construction of antecedents for plural pronouns.

  We are also beginning to develop some insight into the ways in which non-linguistic knowledge can impinge on interpretation. Non-linguistic information can make itself felt either at the level of the lexicon, where it can attach itself to the semantic representations of particular lexical items, or it can provide additional premisses for deductions from the current context DRS.

Syntactic Commitments Is the semantic theory wedded to a particular syntactic theory, or indeed to a particular language?

- Property Theory makes no commitments to a particular syntactic theory, or to a particular language.

  Pen and paper versions of the application of property theory to natural language have made use of categorial grammar. One implemented system uses context free grammar rules with attribute-value features.

- No theory which endeavours to be explicit about the syntax-semantics interface can proceed without making certain syntactic commitments. This has been equally true of DRT as it has been of other theories with comparable aims. At the same time, however, there has been a trend within DRT to play down the dependence of semantics (its primary concern) on any one particular theory of syntax. For instance, [Kamp and Reyle, 1993], arguably the most comprehensive empirical account of DRT to date, goes out of its way to persuade the reader how little the particular syntax it has adopted really matters. The most explicit is this view in the work of Millies and Pinkal [Millies and Pinkal, 1993] which tries to isolate the characteristic semantic constructors of DRT in a setting that is as syntactically neutral as possible.

  But not all DRT-related work is like this. For instance, recent papers by Berman and Hestvik (c.f. [Berman and Hestvik, 1994]) aim to explore the specific features of the interface between DRT and the Theory of Government and Binding. Both within the context of Verbmobil and elsewhere there are now also efforts under way to develop DRT and UDRT on the basis of grammars developed within HPSG (c.f. [Frank and Reyle, 1992], [Frank and Reyle, 1994]).

  LFG-DRT frameworks are reported in e.g. [Reyle, 1985], [Wada and Asher, 1986], [Asher and Wada, 1988] and [Kasper, 1991]. DRT inspired semantic representations have been integrated in unification based categorial grammars [Calder *et al.*, 1988] and [Zeevat *et al.*, 1987]. UDRS construction in "Strictly Lexicalized Grammars" a variant of categorial grammars is described in [König, 1994a].

Inferential Tractability Do the semantic representations lend themselves to computationally tractable inference? Can the representations be obtained and selected in a tractable way? Practical tractability is not the same as low complexity, but rather low complexity for 'reasonable' inputs.

- As Property Theory is first order, it is semi-decidable.

Practical Tractability Practical tractability can be achieved with Property Theory when restricted to finite domains, and also if we have canonical forms for the representations. An example of how the theory can be made practically tractable would be to implement the axioms for propositions and truth as Horn clauses.

Psychological Plausibility Does the semantics (and implied semantic processing) have any psycholinguistic/cognitive plausibility? Does this matter? How do you measure these things anyway?

- We still know so little about the workings of the human mind that it remains virtually impossible to assess how much psychological plausibility any linguistic theory really has - in fact we do not even quite know how the question of psychological plausibility should be phrased.

  This notwithstanding, a try for greater psychological plausibility than seems to attach to formal semantic theories in the tradition of Montague Grammar was certainly one of the original motives behind DRT. A theory of natural language semantics, so the argument went, ought to have something to say also about the ways in which linguistic knowledge is used in the process of language interpretation. And the early successes of the theory (or, anyway, what were perceived to be its early successes) were taken, perhaps overoptimistically, as indications that the information structures the theory posited - the DRSs - as well as the procedures for constructing them out of actual sentences and texts, reflected some aspects of how the human mind structures the information it extracts from the verbal inputs it receives. (It was this conviction also which seemed to lend credence to the idea, central to the DR-theoretical account of attitudes and attitude reports, that DRSs can be fruitfully used as intentional classifiers of attitudinal states.)

  Does psychological plausibility matter? Yes, we think it does, if only because what may appear to be "just" a matter of psychological plausibility when one deals with one part of language is likely to become a matter of plain adequacy, when the account turns to another part, where the expressions of the first part appear within the scope of attitudinal predicates.

Implementation Availability What computational implementations of the theory are available? What would be needed in addition to the semantic theory (and some kind of syntactic theory) to produce an NL system for some inference oriented task like question answering or database access?

- Squirrel is an example of a natural language front end to databases that uses Property-theoretic semantics [De Roeck *et al.*, 1991e]. It is currently being developed as a front end to combined relational databases and textbases. The front end has also been adapted to provide input for a non-monotonic theorem prover [De Roeck *et al.*, 1991b].

  The semantics obtained by the parser are domain independent. It is only when translating the Property-theoretic representation into a relational query that domain specific information is introduced. The domain specific information is stored in an extended data model, which is used to translate the query into a database query language and rule out nonsensical interpretations. When wedded to the theorem prover, a sortal hierarchy was used for a similar purpose. This hierarchy was also used to constrain the statements of the theory to be used in the proof.

- Implementations can be classified (at least) according to the following (somewhat orthogonal) criteria:
  - coverage: here the range is from prototype/toy/research/didactic (i.e. text book type) systems to large scale applications.
  - application area: is the system used for database query, question answering, translation etc.?
  - syntactic commitments: is the implementation in question based on a particular syntactic formalism or theory (like e.g. categorial grammar, phrase structure grammar, LFG, HPSG etc.).
  - mode: does the system support analysis or generation or both (is it reversible etc.)?
  - processing architecture: sequential or simultaneous. In the case of e.g. analysis mode, does the system first perform a syntactic analysis and then performs the construction of semantic representations or are the two processes interleaved (e.g. as instances of one and the same constraint resolution process like in the case of some unification based grammar formalisms)?
  - semantic composition type: is the system based on the DRS-construction algorithm, on a λ-calculus style reformulation, on a unification based approach etc.?

- underspecification: does the system support (some version of) underspecification?
- inferential capabilities: what kind of inferences (if any) are supported by the system? Here the range is from none - fairly limited *linguistic* inferences (etc. antecedent construction and resolution, limited modes of disambiguation etc.) to general reasoning?
- implementation details and other aspects: implementation language, documentation, availability etc.

Below we will briefly present some DRT implementations. In doing so we will follow a roughly chronological perspective and in each case try to comment on the classification criteria which seem most relevant to the system under consideration. While we cannot hope to be comprehensive (both with respect to the number of systems covered and the discussion of implementation classification criteria) we have tried to collect some of the main strands of DRT implementations.

It is striking to note that hardly any of the implementations known to us really implements the DRS-construction algorithm as detailed in [Kamp, 1981b] and [Kamp and Reyle, 1993] to the letter but is either based on a (semi-) compositional λ-calculus style or on a unification based reformulation.[4]

Early work on DRT implementations is reported in [Frey and Reyle, 1983], [Frey *et al.*, 1983], [Reyle, 1985], [Guenthner and Lehmann, 1983] and [Guenthner and Lehmann, 1984]. [Frey and Reyle, 1983] and [Frey *et al.*, 1983] describe a prototype analysis system which constructs a database from a narrative text. The system is based on a Prolog LFG parser which is augmented to construct DRS representations. In [Frey and Reyle, 1983] lexical entries are either associated with atomic conditions or partial DRS (essentially DRS-like structures part of which are Prolog variables) which are propagated up parse tree representations and further instantiated through Prolog unification. In contrast, DRS construction described informally in [Frey *et al.*, 1983] is driven off LFG f-structure representations. A version of this is made formally explicit in [Reyle, 1985] where partial DRSs are represented as λ-abstracts over DRS-like structures like

$$\lambda P.\lambda Q.\langle\emptyset,\{\langle\{v\},\{P(v)\}\rangle \Rightarrow \langle\emptyset,\{Q\}\rangle\}\rangle$$

and semantic composition is defined in terms of a function application principle. In these systems scope ambiguity is accounted for in terms

[4] In fact often on both in the sense that a λ-calculus style reformulation is actually implemented (or often rather approximated c.f. [Pereira and Shieber, 1987]) in terms of unification in a logical programming language.

of the order of application of partial DRS resulting in sets of fully specified representations.[5] Apart from their historical interest, what seems remarkable about these systems is the notion of a partial DRS which in some form or other (i.e. either as a λ-abstract or in terms of variables in a unification based framework) informs much subsequent work on DRT-implementations as well as theoretical work.

The work reported in [Wada and Asher, 1986], [Asher and Wada, 1988] and [Wada, 1990] is based on end extends the work reported in [Frey and Reyle, 1983], [Frey *et al.*, 1983] and [Reyle, 1985]. [Wada and Asher, 1986] presents an implementation which constructs DRSs from LFG f-structure representations while [Asher and Wada, 1988] discusses an extended version of the implementation which integrates interpretive constraints on anaphora resolution. Like their predecessors the implementations are based on the notion of a partial DRS which is unpacked in terms of λ-abstracts over DRSs. [Wada, 1990] shows how DRT based semantic representations together with LFG constraints can handle some of the problems encountered in the translation of anaphoric dependencies from English to Japanese. In standard DRT[6] anaphora resolution is defined in terms of accessibility constraints and compatibility with respect to person and gender specifications. These constraints define "logically" possible antecedents and are "hard" constraints in the sense that violations rule out anaphoric relations as ill-formed. [Asher and Wada, 1988] introduce interpretive constraints which establish a ranking between logically possible antecedents. The ranking is established in terms of the cumulative effect of recency, reiteration, parallelism, grammatical function and salience shifting filters assigning numerical weights to discourse referents in the set of logically possible antecedents. The processing architecture is strictly sequential: a parser determines the constituent structure and f-structure according to an LFG grammar; a DRS constructor maps f-structures into DRSs and a resolution component determines logically possible antecedents and determines a discourse salience and focus ranking.

The notion of a λ-DRS (or partial DRS) has been influential in the development of some constraint and feature-structure based implementations like e.g. in the semantic formalism employed in VERBMOBIL (discussed elsewhere in the present document and in [Bos *et al.*, 1994a]) and the "strongly lexicalized grammar" implementations

[5] This is effectively a version of the Cooper-storage mechanism.

[6] This is for the "core" fragment ([Kamp, 1981b] and chapters 1 and 2 in [Kamp and Reyle, 1993]) which treats pronominal reference to singular NPs.

of [König, 1994a]. One of the motivating factors for this influence has been the criticism that the construction of semantic representations in constraint and feature-structure based formalisms appears to be somewhat unprincipled in the sense that, roughly, the construction is effected in terms of the further instantiation of variables in feature structure representations through unification (sharing of variables etc.) where it is difficult to associate a (sensible) straightforward interpretation with the elements (semantic representation of the daughters and mother in some local tree) involved and the semantic construction (combination) process itself.[7] The λ-based reconstruction (or rather: approximation) of the traditional rule-to-rule setup with functions as semantic representations and function composition as construction principle is sometimes an attempt to salvage some of the formal credibility of the fully interpreted traditional setup in a constraint and feature-structure based environment.

In [Sedogbo, 1988] an analysis mode DRT implementation is presented which is of interest mainly since it features some basic inferential capabilities derived from applying a version of Modus Ponens on DRSs viewed as knowledge representations in the framework of a question-answering application where an answer to a query Q is computed if a (possibly further instantiated) representation of Q can be derived from a database (here the antecedently constructed DRS representation).

A declarative reformulation of DRT which is based on and exploits central features of the logical programming paradigm is developed in [Johnson and Klein, 1986]. Here the notion of a partial DRS is implemented in terms of logical variables[8] and unification and the left-to-right anaphoric dependencies modelled in the DRS-construction algorithm [Kamp, 1981b; Kamp and Reyle, 1993] are explicitly encoded in the grammar in terms of the threading technique from logic programming and updates and tests on the threaded representations. The approach is discussed to some extent in the Syntax-Semantics Interface section in the DRT contribution to FraCaS deliverable D8. The original implementation covers a toy fragment and does not handle scope ambiguity. The approach has been quite influential: the paper by [Covington and Schmitz, 1988] extends the empirical coverage of the original implementation; [Covington *et al.*, 1988] show how DRS like structures obtained from parsing NL input can be compiled into a database consisting of extended Prolog clauses for query and

[7] Often there is no clear distinction between object and meta level uses of variables.

[8] In the guise of path equations.

inference purposes etc.; [Smith, 1990] gives an extended version of this; [Johnson and Kay, 1990] describe a substantial modification of the basic architecture: like [Johnson and Klein, 1986] the approach is based on the threading technique but in [Johnson and Kay, 1990] this is "hidden" in the definition of semantic constructor operators on the one hand and in a Prolog based λ-abstraction approximation of semantic composition in the style of [Pereira and Shieber, 1987]. In addition, the paper provides an abstraction barrier between syntactic and semantic representations in terms of three sets of semantic constructor definitions resulting in Montague grammar, Situation Semantics and Discourse Representation Theory style representations depending on which constructor set definitions are in force. In this approach a single syntactic representation is mapped into a number of semantic representations. Millies and Pinkal [Millies and Pinkal, 1993] develop an alternative approach where an attempt is made to make a single semantic representation (here λ-DRT) as independent as possible from the particulars of the underlying syntactic theory in terms of a set of syntactic construction operators. The [Johnson and Kay, 1990] paper also shows how a Cooper-storage based approach to quantifier scope along the lines presented in [Pereira and Shieber, 1987] can be integrated into the grammar. [Schielen, 1993] has extended this approach to plurality and (recently) temporal phenomena.

In the approaches described by [Johnson and Klein, 1986], [Johnson and Kay, 1990] (and the related approaches described in the section above as well as other implementations not discussed extensively here such as for example the co-description[9] based approach to DRS construction in LFG described by [Kasper, 1991] etc.) syntactic and semantic representations are constructed simultaneously in terms of a general constraint resolution mechanism. This property is one of the hallmarks of so called "sign-based[10]" approaches to which we turn now. It is often argued (or rather: hoped or simply assumed) that simultaneous construction of syntactic and semantic representations offers the advantage that constraints from a variety of sources contrive to rule out certain analyses at the earliest possible stage and thus help cutting down the search space in a derivation. This is in

[9] In the LFG literature a distinction is made between *description by analysis* and *co-description*. Description by analysis refers to a setup where first some level of representation is constructed and then serves as input to the construction of further (levels) of representation (i.e. a sequential architecture) while co-description refers to an integrated approach where levels of description are described and constructed simultaneously in terms of constraints and constraint resolution.

[10] The term derives from the HPSG and unification categorial grammar literature.

contrast to the fact that most large scale implementations (e.g. the CLE) rely on a more traditional sequential processing architecture. The apparent contradiction can be clarified in terms of an example: integrated architectures run the risk of massive duplication of effort in case there is an ambiguity in e.g. the semantic representation part which does not correspond to a parallel syntactic ambiguity. In this case syntactic *and* semantic structures would be multiplied (as in current versions of HPSG). A situation like this would not occur in a sequential paradigm where syntactic processing precedes semantic analysis. It seems to us, however, that the available evidence on this issue is not conclusive for the described scenario presents a worst case for integrated architectures. In particular underspecified representations seem well suited for integrated architectures for they avoid unnecessary multiplication of signs due to alternatives which are local to certain parts of the sign while preserving the advantage of integrated approaches that constraints from a variety of sources can interact and thus rule out certain analyses as early as possible. This is explored in the UDRS-based work reported in [Frank and Reyle, 1992] and [Frank and Reyle, 1994].

Probably the first sign-based approach to DRS construction is reported in the Unification Categorial Grammar (UCG) papers [Zeevat *et al.*, 1987] and [Calder *et al.*, 1988]. UCG and its semantic representation language InL which is based on both DRT and a Davidsonian treatment of verb semantics and has inspired further work on compositional versions of DRT c.f. [Zeevat, 1989]. In [Zeevat *et al.*, 1987] and [Calder *et al.*, 1988] semantic composition (like syntactic and phonological composition) is effected solely in terms of further instantiation of variables in the representation through unification. Recently, [König, 1994a] has integrated underspecified DRSs (UDRSs) in a typed unification based categorial grammar like framework (Strictly Lexicalized Grammars SLGs). [König, 1994b] describes a head-corner generator for UDRS based descriptions in SLG. Complex categories in UCG do essentially the work of subcategorization lists in HPSG (and SLG). [Frank and Reyle, 1992] and [Frank and Reyle, 1994] describe a UDRS based semantic construction in an HPSG framework. This approach is further detailed in the Syntax-semantics Interface section of the DRT contribution in the FraCaS deliverable D8.

The implementations discussed so far all fall under the prototype/research/toy fragment/text book type of system addressing particular (often fundamental) issues in DRT and UDRT implementation. DRS or DRS-like representations (and underspecified versions thereof) figure in larger scale applications. The LEX-project (Linguistics and Logic Based Legal Expert System, 1984–1987, University Tübingen and IBM Scientific Center Heidelberg) involved the construction of an expert system on a section of the German criminal code dealing with traffic violations. To this end it had to represent and process (in particular: draw inferences form) formalisations of linguistic, common sense and legal knowledge. For an introduction see [Lehmann, 1988] and the references cited therein. A DRT variant called InL was used as semantic representation language in the ACORD project which was concerned with natural language and graphics interfaces for the construction and interrogation of knowledge bases (for a general introduction see: [Bes and Guillotin, 1991]). The system architecture includes parsers, generators, a dialogue manager, a knowledge base, a deduction component and a graphics system. DRT-inspired semantic representations (so called E-DRSs, "extended"-DRSs) were used as semantic representations in the LILOG project (for a general reference see [Herzog and Rollinger, 1991]). LILOG involved analysis, generation and inference components for text understanding. In this context mention should be made of an extensive E-DRS based temporal reasoning component developed in the project. In λ-DRT [Bos *et al.*, 1994a] is the semantic representation formalism in VERBMOBIL a speech to speech translation project. λ-DRT is discussed further in the section **Survey of Implementations** in the present deliverable.

Coverage Does the semantic theory have wide coverage (pencil and paper; implemented in a working and tested system)? What kinds of thing can the theory *not* handle? How robust is the theory; is coverage all or nothing or can it be made to degrade gracefully? See also deliverable D9.

- Currently, only a small fragment of natural language has been given a compositional treatment in Property Theory. This fragment consists of the PTQ fragment, plus some coverage of plurals and mass terms, prepositions, donkey sentences, and intersentential anaphora.

  The implementation described above (Squirrel) has a PTQ like coverage with prepositions and questions.

  With rule-based syntax, Property Theory will tend to have an 'all-or-nothing' coverage. In principle, however, the weakness of the theory allows representations of sentences without necessarily having a clear idea of their truth conditions. That is, the theory might degrade gracefully if the syntax has a wider coverage than the semantics.

The theory, in itself, does not say anything about belief revision, nor multi-agent discourse modelling. Defeasible inference is also not addressed in the classical version of Property Theory, as it stands. Finally, you do not get the 'correct' treatment of a phenomenon for free, but this may be no bad thing. For other phenomena, it is quite likely to be largely a question of deploying sufficient resources to implement proposals for their treatment.

9.5 Unified views

As said in subsection 9.3.2 and the previous section under the item of coverage, a discussion and check-list 'practical' linguistic viewpoints on semantic theories would give us an empirical perspective on how semantic theories diverge and, more important from the viewpoint of the FraCaS-project, may be combined and unified. This is what the research and discussion will be focussed on in the second stage of the project.

In the present section we try to give a preliminary view on things to come. The aim is to present succinctly how the different approaches represented within FraCaS deal with a select set of linguistic phenomena. The phenomena have been chosen to exemplify five problem areas. These, in their turn, were picked either because dealing with them requires certain fundamental theoretical decisions, which affect the form of a theory well beyond its application to just that area; or because the theories of FraCaS deal with the area in interestingly different ways. The ultimate goal is to achieve what the section title says: a unified view of the various problems arising in the area for computational semantics and of what might be the best way or ways to tackle them.

The five areas are the following:

1. Tense and Aspect
2. Intensionality, Attitudes and Modality.
3. Ellipsis
4. Indexicality, Deixis and Anaphora.
5. Underspecification, ambiguity and vagueness

As already explained in section 9.2, the first four items belong to the core fragments of different semantic theories. The second item provides key motivations for situation semantics and property theory, while the third and fourth item were of main concern to the initiators of DRT and dynamic semantics. The last item is the ground motivation for the school of monotonic semantics.

This section contains the beginnings of our efforts to come to a "unified view" - a view which all of us can support, or at least live with - on a number of issues which in the course of our work on the deliverables D8 and D9 came to appear to us as points of disagreement of divergence between two or more of the theories represented here.

What is presented here is no more than a first taste of what we would like to accomplish eventually. Only for one of the test issues we have selected are the views of the different approaches worked out in some detail. Otherwise we have had to content ourselves for the time being with a mere reference of the test issues themselves, and a rough explanation of why they were chosen.

Test issue: The analysis of the sentence (example 128 of D5)

(9.3) Smith did not travel by air.

Essential to the semantics of this sentence is that it expresses a relation between a contextually determined reference time and a "negated event". Thus the ways in which the different approaches propose to deal with this sentence is composed of two separate questions:

1. how the approach deals with contextually determined reference times, and
2. the treatment it offers of "negated events"

9.5.1 Contextually Determined Reference Times

DRT favours a treatment that is broadly Reichenbachian: the reference time is represented explicitly as a discourse referent. (In the present case, where the sentence does not contain any constituent which could introduce this discourse referent, it must have been part of the context already.)

Situation Semantics offers a similar treatment, but one which is more general: The analysis of a sentence involves the introduction of a number of parameters.

These parameters are either assigned some contextually determined value or else they are quantified existentially. (This however is a difference between Situation Semantics and DRT which does not manifest itself with regard to the example at hand.)

Dynamic Semantics has also adopted a treatment in the spirit of Reichenbach.

Incorporation of such a treatment within Property Theory should be unproblematic also. However, questions of temporal reference have not been a priority within the Property Theory framework thus far.

Such an analysis can also be incorporated into QLF. It has not been implemented there yet (but of course this is equally true of the other approaches, for which implementation has so far been only partial-to-fragmentary in general.)

9.5.2 The Treatment of "Negative Events"

Situation Semantics offers a uniform treatment for both states and events: Both are situations, though these could be further distinguished into states and events by some discriminating feature. This makes it possible to treat a "negative events" also as a situation of some sort.

DRT adopts a version of Davidsonian Event semantics, but adopts, besides events, states as a separate ontological category. With regard to the situations described by sentences involving negated event verbs such as (9.3) this creates the difficulty that one has to decide whether what such sentences describe is a state or event.[11]

QLF offers a uniform Davidsonian treatment of states and events, but has not yet put in place a specific implementation of event anaphora, which in the case of (9.3) seems to be interacting with the state-event distinction in a crucial way.

A similar "ontological neutrality" with regard to the distinction between states and events can be found within Dynamic Semantics and Property Theory.

A further problem is offered by the variant (9.4) of (9.3).

(9.4) Smith did not travel by air, but by train.

[11] C.f. recent work by de Swart.

This sentence shows more clearly than (9.3) itself that such negated event sentences allow for two different construals. Thus (9.3) can be understood either as saying (i) that no event of Smith travelling by air took place, or (ii) that the travelling event in question wasn't by air. Note that these two interpretations are logically independent. (i) does not entail that there was a travelling event, whereas (ii) does not exclude the possibility of there having been besides the event of Smith travelling not by air, another event of his travelling by air. Though (9.4) can be analyzed employing either of these two interpretations for (9.3) - according to (i) (9.4) is elliptical, and "after reconstruction" says that there was no event of Smith travelling by air, but there was an event of him travelling by train. According to (ii) the analysis is that the travelling event in question was one where Smith went by train and not by air. The second analysis, which involves what in the linguistic literature is often referred to as "constituent negation" - here the negation applies to only part of a sentence leaving the remainder of it unaffected. The exemplary cases of constituent negation are precisely sentences of the pattern of (9.4), where a following truncated but-clause provides the material by which the negated constituent should be replaced in order to arrive at the proposition which the speaker endorses.

Chapter 10

Annotated Text

10.1 Introduction

In this section, a text chosen from the European SemEval evaluation is given an informal semantic annotation. The annotation is intended to indicate the kinds of problems confronting computational semantics, and the extent to which the current state of the art is capable of dealing with them.

The section contains:

- the text itself (Financial Times ...)
- a surface oriented syntactic analysis of the first sentence with indexed constituents
- an exemplary "in depth" analysis of the first sentence, which is intended to give an impression fo the semantic complexity of real natural language texts
- a detailed but somewhat less fine-grained analysis of the whole text.

10.2 A Sample Text

The text in question is the following:

Financial Times of 28 July 1994, page 19

Euro Disney cuts its operating loss despite revenue fall

by John Ridding in Paris

Euro Disney, the troubled leisure group, announced a net loss of FFr546m (sterling 66m) for the third quarter to the end of June, but said that operating losses had been reduced sharply from FFr381m to FFr194m.

The increase in losses at the net level was the result of a one-off charge of FFr352m relating to a financial restructuring at the company, which is implementing a rescue package after suffering losses of FFr5.3bn last year.

The reduction in operating losses was achieved despite a fall in revenues at the theme park and its hotels from FFr1.47bn in the third quarter of 1993 to FFr1.16bn this year. Euro Disney blamed a more aggressive pricing strategy and lower attendances after uncertainty last year over the future of the theme park.

"Some people outside of the company were talking of the possibility that we might have to close," it said. "This was never justified , but had an unsettling effect on clients and potential customers"

10.3 A Syntactic Analysis

$[_{S1}$ $[_{NP1.2}$ $[_{NP1}$ ***Euro Disney***$_{N1}]$

$[_{NP2}$ ***the***$_{Det1}$

$[_{N'2}$ ***troubled***$_{A1}$

$[_{N2}$ ***leisure***$_{N3}$

group$_{N4}]]]]$

$[_{VP1}$ ***announced***$_{V1}$

$[_{NP5}$ ***a***$_{Det2}$

$[_{N'5}$ ***net***$_{A2}$

loss$_{N5}$

$[_{PP1}$ ***of***$_{P1}$

$[_{NP6}$ ***FFr546m***$_{N6}]$

$[_{PP2}$ ***for***$_{P2}$

$[_{NP7}$ ***the***$_{Det3}$

$[_{N'7}$ ***third***$_{A3}$

$\boldsymbol{quarter}_{N7}]$
$[_{PP3}\ \boldsymbol{to}_{P3}$
$\quad [_{NP8}\ \boldsymbol{the}_{Det4}$
$\qquad [_{N'8}\ \boldsymbol{end}_{N8}$
$\qquad\quad [_{PP4}\ \boldsymbol{of}_{P4}$
$\qquad\qquad [_{NP9}\ \boldsymbol{June}_{N9}]]]]]]]]]]]$

10.4 In Depth Description

NP1/N1: ***Euro Disney:*** proper noun (syntactically complex name, semantically atomic); implies unique reference in the context of utterance.

Options:

1. no such reference object is available: introduce (accommodate) new reference object, attach information that it is named "Euro Disney"
2. reference object named "Euro Disney" available in context (typically: world knowledge, permanent memory): identify $NP1$ referent with it. Problem: Ambiguity by "layered ontology": $NP1$ may alternatively refer to (at least:) physical object (amusement park near Paris), group of people (Euro Disney employees), or institution (business organisation).

N4 ***group*** amb.: (R1: collection of objects/persons, R2: an institution formed by such a collection); domain-specific specification of R2: commercial institution formed by a group of companies

N3 ***leisure*** means in general something like "pleasurable activity" (difficult ontology); domain-specific reading: a branch of industry providing leisure facilities

N2: ***leisure group***, compound Nominal: in general large potential for ambiguity, due to underspecified relation between constituents. One possible reading: "a group indulging in pleasurable activities". Plausible domain/specific reading: "a business organisation providing services in the leisure industry"

A1 ***troubled:*** adjective, intersective (?), sortal requirement for CN it combines with: individuals that can be troubled; difficult ontology

N'2 ***troubled leisure group:*** standard modification construction: expressing the property of

1. being troubled and
2. being a leisure group

NP2: the determiner ***the*** makes the NP definite, implying uniqueness of a referent satisfying the description $N'2$ (or at least existence of a sufficiently salient referent). There are in general two options (as in the case of $NP1$, which is likewise a definite noun phrase):

1. A referent is found in the context and identified with the referent of $NP2$, or:
2. a new referent is accommodated and assigned the property of being a troubled leisure group.

The special syntactic context of $NP2$ triggers yet another semantic treatment:

NP1.2 is an appositive construction, which forces identification of the referents of the head $NP1$ and the non-restrictive modifier $NP2$. Several things result from this fact:

1. As in non-restrictive constructions in general, a unique referent is given for $NP2$, the descriptive content of $NP2$ is accommodated (if not contained in the data-base before)
2. The institution reading for ***group*** is selected, since the NP refers to a named entity.
3. This specifies ***Euro Disney*** to refer to the institution level.

V1 ***announced:*** transitive action verb expressing a communicative act of a certain formal status, a natural person or institution being the subject, and an event or proposition the direct object, and "the public" the (usually implicit) default goal of the communication. ***announced*** is a simple past tense form: Temporal location of the event expressed by the sentence in the past, at a contextually given reference time. Domain-specific use: event has taken place at some time within some recent period (the report time span of the present newspaper copy).

N5 ***loss:*** a deverbal abstract noun with an internal argument position (theme of "lose"), expressing a state change: The loss of x expresses the change from having x to not having x. If the argument is measure term, it may also express in a more abstract reading, namely the numeric difference between former and later state with respect to the dimension specified by the measure unit.

NP6 ***FFr546m:*** measure NP with "FFr" expressing dimension and unit of measurement, and 546m expressing the amount; as in all measure terms, ambiguity between "exactly" and "at least" reading.

PP1 ***of FFr546m:*** preposition ***of*** semantically empty, marks NP6 as argument of $V1$.

A2 ***net:*** an adjective which applies to common nouns expressing measurable units. Like "total" or "precise" or "average", it functions as a linguistic hedge, enforcing a specific reading of the noun/N' it modifies. As one effect, it clearly induces the "exactly" reading of the amount. However, there are other components in its semantics which can be defined only by referring to domain-specific knowledge (finances).

NP5/Det2 ***a net loss of FFr546m ...:*** def/indef is of no importance, because the loss is mentioned for the first time but is unique.

N7 ***quarter:*** one standard reading (selected by domain) is "quarter of a year"; functions as head in partitive constructions. The default argument is the current year.

A3 ***third:*** ordinal number; functional semantics: selects the third element

N'7 ***third quarter:*** requires specification what basis is assumed for counting quarters; calendar year / financial year .

NP7 ***the third quarter ...*** definite NP, uniqueness structurally given by the functional meaning of ***third***.

PP2 ***for the third quarter ...*** individuates the event $NP5$ refers to by specifying a time-interval.

NP9/N9 ***June*** is a proper name which is indexical: context must provide the year.

N8 ***end:*** relational noun with internal argument; if the argument is a temporal interval, ***end*** maps it to its endpoints.

NP8 ***the end of June*** definite NP, uniqueness structurally given by the functional meaning of ***end.***

PP3 ***to the end of June*** temporal PP, partially locates $N'7$: The goal point of the third quarter is the end of June. Hereby resolving the ambiguity of $N'7$ to "the third quarter of the current financial year". The PP semantically functions as an apposition to $NP7$ and should possibly be adjoined to it.

10.5 Annotation

Comments / annotations of the text are divided into three broad categories

1. *Description:* Describes a semantic (or sometimes, syntactic) phenomenon of interest.
2. *Ambiguity:* There is more than one possible interpretation of the phrase, and some appeal to context is typically required to resolve the ambiguity
3. *Open Problem:* As the name suggests, a phenomenon that poses particular difficulties for the current state of the art. One should not assume, however, that everything listed under 'description' or 'ambiguity' is unproblematic.

Financial Times of 28 July 1994, page 19

— *Description:* Elliptical title, no linguistic antecedent: i.e. This text is taken from... (Can't be construed as uniquely identifying name of text)

— *Description:* Meaning of "of": in this case, means "printed on (date)"

— *Description:* Meaning of comma: page 19 of the FT

— *Description:* Which indicates that "page" is being used as a relational noun

Euro Disney

— *Description:* Complex name with some descriptive content through internal structure

cuts

— *Open Problem:* Headline present tense to refer to past event (Euro Disney has cut...)

its

— *Ambiguity:* Possessive pronoun referring to Euro Disney

operating loss

— *Ambiguity:* Adjectival gerund: financial loss incurred through operation, not loss of operations or deaths through performing surgery…

despite

— *Open Problem:* Connective taking nominalised event as complement (can take other NP complements as well)

revenue fall

— *Description:* Missing determiner

— *Description:* Compound nominal

— *Description:* Lexical semantics: “fall” as noun and verb — from existence of a revenue fall, we can infer that the revenue fell.

by John Ridding in Paris

— *Description:* Elliptical title, no linguistic antecedent: i.e. This report was written by John Ridding in Paris

Euro Disney, the troubled leisure group,

— *Description:* Appositive nominal modification; does the singular definite determiner on its own imply any kind of contextual uniqueness? This is arguable in appositive contexts.

— *Ambiguity:* Compound nominal (leisure group): in this context means a business organisation/group providing leisure facilities, not a group indulging in pleasurable activities

— *Description:* Verbal adjective (troubled): adjective derived from (passive?) verb: is there a missing argument for what is troubling Euro Disney?

announced

— *Description:* Argument coercion: do individuals and organisations make announcements in the same way? [answer depends on application]

— *Description:* Announce <NP>: NP either has to be event denoting, or elliptical e.g. 'announced that they had/made a loss of..'

a net loss of FFr546m

— *Description:* Net loss is accumulation of all other losses, so

— *Ambiguity:* FFr546m cannot be distributive with wide scope over "a loss"

— *Ambiguity:* Scalar implicatures cancelled?: can you have a net loss of of FFr546m if in fact you have lost more?

— *Ambiguity:* Is 546m is determiner/measure term? [yes]. If so, do we have a a particular 546m francs in mind — clearly not.

— *Description:* Loss in financial sense: Euro Disney has not mislaid some money and is now searching for where it left it.

(sterling 66m)

— *Description:* Parenthetical appositive: FFr546m = sterling 66m

for the third quarter

— *Description:* A net loss for the third quarter accumulates losses in the third quarter

— *Ambiguity:* Financial sense of quarter (presupposes domain knowledge for the finance domain)

to the end of June,

— *Ambiguity:* Restrictive or non-restrictive modification to "third quarter" (to be resolved on the basis of domain knowledge: does the financial quarter normally run to end of June?)

— *Ambiguity:* "to the end" = "lasting to the end"

— *Ambiguity:* June: needs resolution to June 1994

but

— *Ambiguity:* Contrastive sentential connective: implies that net loss has increased, in contrast to fall in operating losses

said

— *Description:* As with announcements, how do organisations say things (depends on domain)

that operating losses

— *Open Problem:* Losses in the plural: criteria for individuating and counting losses? (Just because there are no clear criteria for that the distinction between singular and plural has no semantic relevance here)

had been reduced

— *Description:* Indirect speech / sequence of tense: Euro Disney say "losses have been reduced"

— *Ambiguity:* It is the total size of the losses that has been reduced, not the number of individual losses, or the sizes of individual losses

sharply from FFr381m to FFr194m.

— *Description:* "Sharply" is an adverb of manner, qualifying the reduction

The increase in losses at the net level

— *Open Problem:* Definite reference to the increase: antecedent only introduced either by use of "but" in preceding sentence, or (perhaps) through financial knowledge that eg. net loss = (revenue - operating costs) and reference to headline: but this is contradicted by reference below to financial restructuring. This may be a case of 'reportive definite', where new information is crammed into the description for concise reporting.

— *Ambiguity:* Does "at the net level" attach to "increase", or can you have more than one loss at the net level? (Domain knowledge to the effect that 'loss(es) at the net level' is cumulative resolves this.

— *Description:* “Increase”: relation between verb and noun.

was the result

— *Description:* Singular definite: but presumably the increased loss was not the only result of the restructuring. Definiteness in context: we are focussing on the financial effect of the restructuring here.

of a one-off charge

— *Open Problem:* Is it the charge itself, or making the charge that results in the increase? Can there be a charge without the corresponding event of making the charge? The distinction may seem scholastic here: the charge is the event of making the charge itself. But to account for this we need a more subtle theory of events than we have at present.

— *Ambiguity:* “one-off”: what is a non one-off charge? A charge that can have different tokens of the same type? A function from times or situations to charges? Presumably, charges are identified by their headings in the account book: ‘bribes’, ‘salaries’, ‘taxes’, etc. A one-off charge is one with a heading that occurs only once, or at least is expected to occur only once, e.g. ‘restructuring of organization’ (assuming that the organization is only restructured once). Again: reasoning on the basis of domain knowledge should establish that this is what is intended. The criteria of identification of domain specific concepts like ‘charge’ are to be provided by domain knowledge.

— *Ambiguity:* “one-off”: one-off over some contextually given period, or for all time?

of FFr352m relating

— *Ambiguity:* Adjectival gerund: what kind of relation? (but maybe the relation could be left underspecified here). Modifies “charge” not FFr

to a financial restructuring

— *Description:* Gerund nominalisation

at the company,

— *Ambiguity:* "at the company" = "of the company" or "of part of the company"

which

— *Description:* Non-restrictive relative clause, modifiying company, not restructuring

is implementing a rescue package after suffering losses

— *Ambiguity:* Temporal connective with non-finite VP complement (not another gerund nominalisation). Implicit subject is EuroDisney.

of FFr5.3bn last year.

— *Ambiguity:* Last year = 1993

The reduction

— *Ambiguity:* Definite NP with antecedent inferred from preceding VP

— *Description:* Event nominalisation

in operating losses

— *Ambiguity:* Reduction measured in terms of cumulative size, not number of losses (requires domain knowledge, reasoning)

was achieved despite a fall in revenues

> — *Ambiguity:* Plural revenues: Why? Revenue for park and revenues for hotels? Again, the singular/plural distinction may not be semantically relevant here, just because the individuation criteria for ‘revenue’ are so sloppy. Same situation as with loss/losses. The sense of the sentence would not change were the plural replaced by a singular.

at the theme park

> — *Ambiguity:* Functional definite: world knowledge that EuroDisney runs a a theme park

> — *Description:* “Theme park”: compound nominal; what is the relation between theme and park?

and

> — *Description:* NP conjunction

its hotels

> — *Ambiguity:* its = EuroDisney’s or the theme park’s??

from FFr1.47bn in the third quarter

> — *Ambiguity:* “from...” modifies “fall” but the amount refers to revenues, and “the third quarter” also specifies the period of the revenue.

of 1993 to FFr1.16bn this year.

> — *Ambiguity:* More temporal anaphora: 1994

Euro Disney blamed a more aggressive pricing strategy and lower attendances

— *Description:* NP conjunction

— *Description:* Comparative NPs

— *Ambiguity:* More agressive and lower than what: than the same things last year?

— *Ambiguity:* What is an aggressive pricing strategy: lowering prices (aggression against competition) or rasing prices (aggression against customers)

— *Ambiguity:* Pricing strategy and attendances are for theme park and hotel, but this is left implicit

— *Ambiguity:* Also left implicit is what they are to blame for.

after

— *Ambiguity:* Implied causal link or not? Uncertainty led to pricing strategy and low attendances, or just a coincidental succession.

— *Ambiguity:* Elliptical complement: after there was uncertainty last year

uncertainty

— *Ambiguity:* Whose uncertainty? Maybe should be left underspecified. Depends on whether there is implied causal link.

last year

— *Ambiguity:* Temporal anaphor.

over the future

— *Ambiguity:* Functional definite: but why do theme parks have a future while other objects do not (e.g. pebbles, snowflakes). Due to omain knowledge, and has also something to do with almost metaphorical treatment of things or organisations as if they were living beings.

of the theme park.

"Some people outside of the company

— *Ambiguity:* Definite anaphor, depending on knowledge that EuroDisney is a company

were

— *Description:* Temporal anaphor: when were they talking? Last year?

talking

— *Description:* Talking does not induce existential commitment, i.e. you can talk of a possibility without that possibility existing, or maybe existence of a possibility does not ammount to much anyway.

of the possibility

— *Open Problem:* Reification of possibilities; or is this just a possible event?

that we might have to close,"

— *Description:* Combining different kinds of modal: epistemic and deontic/root

it said.

— *Description:* Direct speech (How does EuroDisney speak?) Metaphoric talk about organisations as living beings; this is language specific, and the rules in German are more strict.

“This

— *Ambiguity:* Anaphoric: this = the talk

was never

— *Description:* Adverb of quantification

justified, but

— *Description:* VP conjunction

had an unsettling effect on clients and potential customers”

— *Ambiguity:* Scope: one effect on all clients and customers, or slightly different effects?

— *Description:* Non-intersective adjective: potential customers aren’t necessarily customers

Chapter 11

What can't yet be done

11.1 Problems for the future

In this section we give a brief description of some aspects of natural language semantics that currently cannot be handled but need to be addressed as research topics in the near future.

11.1.1 Contextual Reasoning

At the current state fo the art "deep" contextual reasoning (i.e. involving deduction or abduction) is only possible for small domains possessing a clear logical or formal structure. The effort of developing and maintaining such descriptions is large even for these small domains and only a few research groups world-wide have any degree of practical experience in this area. Implementations are characteristically slow and computationally intensive, since the amount of reasoning involved typically requires searching through a large problem space.

"Shallow" contextual reasoning (e.g. some types of probabilistic disambiguation) is possible for much wider domains, given the appropriate training material, but applies to a smaller set of problems. In general, statistical method in contextual reasoning are used to select the most likely interpretation from some candidate set. There are fewer cases in which statistical methods can be used to actually *generate* such candidates for selection.

To some extent, the situation in contextual reasoning is analogous to that in syntax and parsing (although the analogy should not be pressed too far, for in contextual reasoning neither theory nor practice are as well developed). Complete syntactic analyses are only reliably obtainable for short to medium length sentences (circa 15 words), in "clear" corpora. Shallow phrase level parsing is possible for (almost) any text, via statistical methods (or hybrid statistical and symbolic methods). Of course, such an analysis is less informative than a complete one, and may be only partly accurate.

At present, both for parsing and contextual reasoning, the choice is between deep analysis and narrow coverage, or shallow analysis and wide coverage. What is urgently needed, in both areas, is some way of bringing the two strategies together, so that within the same sentence, those units that are capable of full analysis receive that level of analysis, while the rest of the sentence receives an analysis on whatever level is possible. In order to achieve this "variable depth" style of analysis it will be necessary to develop statistical or other robust methods both for the generation of alternative candidate interpretations and for the selection of the preferred one with respect to the current context. It will also be necessary to develop further current ideas about "underspecified" representations so as to allow for a uniform way of representing different depths of analysis. Reasoning techniques capable of using such representations are also required.

11.1.2 Spoken language input

The large-scale commercial success of NL systems crucially depends on the availability of flexible speech front-ends. Recognition systems for spontaneous speech will become available during the next years. They will impose strong additional requirements on linguistic processing components - particularly on systems for semantic processing.

Currently, there exist two fundamentally different sorts of systems for semantic processing. First, purely statistical systems for semantic processing which are doomed to failure, except for very simple and restricted applications. Second, knowledge-based methods, which rely on deep modelling of linguistic and conceptual structure; they are as yet plainly inapplicable to realistic spoken-language processing tasks. In particular, they are unequipped to deal with several kinds of radical underspecification found in spoken discourse.

Spontaneous-speech utterances may be incomplete ("elliptic" in an extreme way), incoherent (consisting of a syntactically unconnected sequence of phrases) and inconsistent (grammatically wrong in various ways). The presence of one or more of these properties is the default case rather than an exception, in informal discourse. Therefore, semantic analysis must be able to process incomplete and incoherent input information. There will be no fully specified syntactic basis for conventional compositional analysis. With regard to inconsistent utterances, these do not pass current speech recognition systems which enforce a grammatically correct structure on utterances through their language model. However, this structure may simply be wrong, given the context and speaker intentions. In the long run, ways must be found to deal with inconsistent input as well.

Further problems for semantic processing arise from uncertainty. Speech recognition systems do not provide one single correct analysis of the spoken input. Rather, they supply either an ordered and weighted list of sentence hypotheses (n-best) or a word lattice consisting of various alternative and overlapping word hypotheses. The performance of available speech recognition systems does by no means guarantee that the correct analysis is found among the highest-rated hypotheses. As this situation will not change during the next years, at least for spontaneous speech recognizers, semantic analysis must be able to work with highly uncertain, in many cases wrong or misleading input information, pursue many alternatives in parallel, make choices and change weights, where this is needed.

Thus, one of the most urgent tasks in NL processing is to develop computational semantics in a way that it meets the requirements imposed by radically underspecified, inconsistent and uncertain information. To achieve this goal, representation formalisms for knowledge-based semantic processing must be further developed, and knowledge-based and statistical methods have to be integrated in an appropriate way.

11.1.3 Dialogue

Most of the successful work in semantics over the past twenty years or so which has applications to natural language technology has involved sentence and text processing. However, it will be important in the immediate future to develop formal models of communication in dialogue that incorporate the results of formal semantics at the same time as making explicit how the particular problems of communicating agents are to be treated. The central problems on which progress might be made in the near future include:

- An adequate characterization of agents' information states appropriate to the treatment of dialogue.
- An account of updates of agents' information states in relation to dialogue games.
- A realistic treatment of communication in dialogue that takes into account the possibility that communication might be quite successful for the purposes at hand without there being an exact match between the information states of the dialogue participants. A practical model of communication in dialogue should be able to say something about how the participants' information states may approximate each other and the associated degrees of successful communication.
- A treatment of context that takes into account that in dialogue the reliance on information about the context becomes of even more central importance than in text interpretation, in particular shared or shareable information about the visual and otherwise perceivable environment of the dialogue participants. Formal models which integrate visual information (and other information gained by perception) with linguistic semantic information need to be built.

We need to construct rigorous models of information exchange in dialogue which take into account central phenomena of dialogue such as information updates by declarative utterances, the role of goals in questions and answers, ellipsis resolution, negotiation of meaning, and the role of focus phenomena in dialogue games.

Bibliography

[Aczel, 1980] Aczel, P. 1980. Frege structures and the notions of proposition, truth and set. In Barwise, J.; Keisler, H.J.; and Kunen, K., editors 1980, *The Kleene Symposium.* North-Holland, Amsterdam. 31–59.

[Aczel, 1988] Aczel, P. 1988. *Non-Well-Founded Sets.* CSLI Lecture Notes. CSLI, Stanford.

[Allen, 1987] Allen, J. F. 1987. *Natural Language Understanding.* Benjamin Cummings, Menlo Park, CA.

[Alshawi and Carter,] Alshawi, H. and Carter, D. ter. Training and scaling preference functions for disambiguation. *Computational Linguistics, forthcoming.*

[Alshawi and Crouch, 1992] Alshawi, H. and Crouch, R. 1992. Monotonic semantic interpretation. In *Proceedings 30th Annual Meeting of the Association for Computational Linguistics.* 32–38.

[Alshawi and van Eijck,] Alshawi, H. and Eijck, J.van jck. Logical forms in the Core Language Engine.

[Alshawi, 1987] Alshawi, H. 1987. *Memory and Context for Language Interpretation.* Cambridge University Press, Cambridge, England.

[Alshawi, 1990] Alshawi, H. 1990. Resolving quasi logical form. *Computational Linguistics* 16:133–144.

[Alshawi, 1992] Alshawi, H., editor 1992. *The Core Language Engine.* MIT Press, Cambridge Mass, Cambridge, Mass., and London, England.

[Appelt and Hobbs, 1990] Appelt, Douglas and Hobbs, Jerry 1990. Making abduction more efficient. In *Proceedings, Darpa Speech and Natural Language Workshop, Hidden Valley.* Morgan Kaufmann, Inc., San Mateo, California. 60–62.

[Asher and H.Kamp, 1989] Asher, N. and H.Kamp, 1989. Self-reference, attitudes and paradox. In Chierchia, Gennaro; Partee, Barbara; and Turner, Raymond, editors 1989, *Properties, Types and Meaning*, volume 1. Kluwer, Dordrecht.

[Asher and Wada, 1988] Asher, N. and Wada, H. 1988. A computational account of syntactic, semantic and discourse principles for anaphora resolution. *Journal of Semantics* 6(3):309–344.

[Asher, 1993] Asher, N. 1993. *Reference to Abstract Objects in Discourse.* Kluwer, Dordrecht, Dordrecht.

[Barros and De Roeck, 1994] Barros, Flávia and De Roeck, Anne 1994. Resolving anaphora in a portable natural language front end to a database. In *Proceedings of the 4th ACL Conference on Applied Natural Language Processing*, Stuttgart.

[Barwise and Cooper, 1993] Barwise, J. and Cooper, R. 1993. Extended Kamp notation. In Aczel, P.; Israel, D.; Katagiri, Y.; and Peters, S., editors 1993, *Situation Theory and its Applications, v.3.* CSLI. chapter 2, 29–54.

[Barwise and Etchemendy, 1987] Barwise, J. and Etchemendy, J. 1987. *The Liar.* Chicago University Press, Chicago, Chicago.

[Barwise and Perry, 1983] Barwise, J. and Perry, J. 1983. *Situations and Attitudes.* MIT Press, Cambridge Mass, Cambridge, MA.

[Barwise, 1981] Barwise, J. 1981. Scenes and other situations. *The Journal of Philosophy* 78:369–397.

[Barwise, 1987] Barwise, J. 1987. Noun phrases, generalized quantifiers and anaphora. In Gärdenfors, P., editor 1987, *Generalized Quantifiers: linguistic and logical approaches.* Reidel, Dordrecht. 1–30.

[Bealer, 1982] Bealer, G. 1982. *Quality and Concept.* Clarendon Press, Oxford, Oxford.

[Benthem *et al.*, February 1993] Benthem, J. van; Eijck, J.van; and Frolova, A. 1993. Changing preferences. Technical Report CS-R9310, CWI.

[Berman and Hestvik, 1994] Berman, S. and Hestvik, A. 1994. Principle b, drt and plural pronouns. to appear in: Arbeitspapiere des sonderforschungsbereichs 340, IMS Stuttgart, Germany.

[Bes and Guillotin, 1991] Bes, G. G. and Guillotin, T., editors 1991. *A Natural Language and Graphics Interface.* Springer Verlag, Berlin, Germany.

[Bierwisch, 1983] Bierwisch, M. 1983. Semantische und konzeptuelle repräsentationen lexikalischer einheiten. In Ruzicka, R. and Motsch, W., editors 1983, *Untersuchungen zur Semantik, Studia Grammatika 22.* Akademie Verlag, Berlin. 61–99.

[Boguraev and Briscoe, 1988] Boguraev, B. and Briscoe, E., editors 1988. *Computational Lexicography or Natural Language Processing.* Longman, Harlow, Essex.

[Bos *et al.*, 1994a] Bos, J.; Mastenbroek, E.; McGlashan, S.; Millies, S.; and Pinkal, M. 1994a. The verbmobil semantic formalism. VM-Report 6, Universität des Saarlandes.

[Bos *et al.*, 1994b] Bos, J.; Mastenbroek, E.; McGlashan, S.; Millies, S.; and Pinkal, M. 1994b. A compositional DRS-based formalism for NLP-applications. In *International Workshop on Computational Semantics*, Tilburg.

[Briscoe *et al.*, 1993] Briscoe, E.; Paiva, V.de; and Copestake, A., editors 1993. *Inheritance, Defaults and the Lexicon.* Cambridge University Press.

[Calder *et al.*, 1988] Calder, J.; Klein, E.; and Zeevat, H. 1988. Unification categorial grammar. a concise, extendable grammar for natural language processing. In *COLING 88, Conference Proceedings*, Budapest. 83–86.

[Charniak and Goldman, 1988] Charniak, E. and Goldman, R. 1988. A logic for semantic interpretation. In *Proc 26th ACL, Buffalo.* Assoc for Comp Ling. 87–94.

[Chomsky, 1981] Chomsky, N. 1981. *Lectures on Government and Binding.* Foris, Dordrecht.

[Cooper, 1975] Cooper, R. 1975. *Montague's Semantic Theory and Transformational Syntax.* Ph.D. Dissertation, University of Massachusetts, Amherst.

[Cooper, 1983] Cooper, R. 1983. *Quantification and Syntactic Theory.* Reidel, Dordrecht, Dordrecht.

[Copestake, 1990] Copestake, A. 1990. An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary. In *Workshop on Inheritance in Natural Language Processing, Proceedings.* Tilburg. 19–29.

[Copestake, 1993] Copestake, A. 1993. Defaults in lexical representation. In Briscoe, E.; Paiva, V.de; and Copestake, A., editors 1993, *Inheritance, Defaults and the Lexicon.* Cambridge University Press.

[Covington and Schmitz, 1988] Covington, M. and Schmitz, N. 1988. An implementation of discourse representation theory. Artificial Intelligence Research Report 01-0023, The University of Georgia, Athens, Georgia.

[Covington *et al.*, 1988] Covington, M.; Nute, D.; Schmitz, N.; and Goodman, D. 1988. From english to prolog via discourse representation theory. Artificial Intelligence Research Report 01-0024, The University of Georgia, Athens, Georgia.

[Dalrymple *et al.*, 1993] Dalrymple, M.; Lamping, J.; and Saraswat, V. 1993. Lfg semantics via constraints. In *Proceedings of the 6th Meeting of the European ACL (EACL)*, Utrecht.

[DARPA, 1989] DARPA, 1989. *Proceedings, Darpa Speech and Natural Language Workshop, Cape Cod.* Morgan Kaufmann, Inc., San Mateo, California.

[De Roeck *et al.*, 1991a] De Roeck, A.; Ball, R.; Brown, E.K.; Fox, C.J.; Groefsema, M.; Obeid, N.; and Turner, R. 1991a. Helpful answers to modal and hypothetical questions. In *Proceedings of the European ACL 1991*, Berlin.

[De Roeck *et al.*, 1991b] De Roeck, A.; Ball, R.; Brown, E.K.; Fox, C.J.; Groefsema, M.; Obeid, N.; and Turner, R. 1991b. Helpful answers to modal and hypothetical questions. In *Proceedings of the European ACL 1991*, Berlin.

[De Roeck *et al.*, 1991c] De Roeck, A.N.; Fox, C.J.; Lowden, B.G.T.; Turner, R.; and Walls, B. 1991c. A formal approach to translating English into SQL. In *Aspects of Databases (Proceedings of the Ninth British Conference on Databases)*, Wolverhampton. Butterworth-Heinmann.

[De Roeck *et al.*, 1991d] De Roeck, A.N.; Fox, C.J.; Lowden, B.G.T.; Turner, R.; and Walls, B. 1991d. A natural language system based on formal semantics. In *Proceedings of the International Conference on Current Issues in Computational Linguistics*, Penang, Malaysia.

[De Roeck *et al.*, 1991e] De Roeck, A.N.; Fox, C.J.; Lowden, B.G.T.; Turner, R.; and Walls, B. 1991e. A formal approach to translating English into SQL. In *Aspects of Databases (Proceedings of the Ninth British Conference on Databases)*, Wolverhampton. Butterworth-Heinmann.

[Eijck, 1994] Eijck, J. van 1994. Presupposition failure — a comedy of errors. *Formal Aspects of Computing* 6:1–22.

[Elworthy, 1993] Elworthy, D. 1993. *The Semantics of Noun Phrase Anaphora.* Ph.D. Dissertation, University of Cambridge, Darwin College, Cambridge, U.K.

[Enderton, 1972] Enderton, H.B. 1972. *A Mathematical Introduction to Logic.* Academic Press, New York.

[Escriba, 1992] Escriba, J.B. 1992. Two formal systems for situation semantics. *Notre Dame Journal of Formal Logic* 33:70–88.

[Fenstad *et al.*, 1987a] Fenstad, F.; Halvorsen, P.K.; Langholm, T.; and Benthem, J. van 1987a. *Situations, Language and Logic.* D. Reidel, Dordrecht, The Netherlands.

[Fenstad *et al.*, 1987b] Fenstad, J.E.; Halvorsen, P.K.; Langholm, T.; and Benthem, J.van 1987b. *Situations, Language and Logic.* D.Reidel, Dordrecht.

[Fiengo and May, 1991] Fiengo, R. and May, R. 1991. *Indices and Identity.* draft.

[Fitting, 1990] Fitting, M. 1990. *First-order Logic and Automated Theorem Proving.* Springer Verlag, Berlin.

[Fox, 1993] Fox, C.J. 1993. *Plurals and Mass Terms in Property Theory.* Ph.D. Dissertation, University of Essex, Colchester, U.K.

[Fox, 1994] Fox, Chris 1994. Discourse representation, type theory and property theory. In *International Workshop on Computational Semantics, Tilburg, ITK, December 1994.* 71–80.

[Frank and Reyle, 1992] Frank, A. and Reyle, U. 1992. How to cope with scrambling and scope. In Görtz, G., editor 1992, *CONVENS 92.* Springer, Berlin.

[Frank and Reyle, 1994] Frank, A. and Reyle, U. 1994. Principle based semantics for hpsg. to appear in: Arbeitspapiere des sonderforschungsbereichs 340, IMS Stuttgart, Germany.

[Frey and Reyle, 1983] Frey, W. and Reyle, U. 1983. A prolog implementation of lexical functional grammar as a base for a natural language processing system. In *EACL 83, Pisa, Italy.* 52–56.

[Frey *et al.*, 1983] Frey, W.; Reyle, U.; and Rohrer, C. 1983. Automatic construction of a knowledge base by analysing texts in natural language. In *IJCAI 83, Karlsruhe, Germany.*

[Frey, 1993] Frey, W. 1993. *Syntaktische Bedingungen für die semantische Interpretation.* studia grammatica. Akademie Verlag, Berlin.

[Gardent, 1991] Gardent, C. 1991. *Gapping and VP Ellipsis in a Unification-Based Grammar.* Ph.D. Dissertation, Univ. of Edinburgh.

[Geurts, 1994] Geurts, 1994. *Presupposition.* Ph.D. Dissertation, Universität Stuttgart, Stuttgart, Germany.

[Gillon, 1990] Gillon, B. 1990. Ambiguity, generality, and indeterminacy: Tests and definitions. *Synthese* 85:391–416.

[Girard, 1987] Girard, J.Y. 1987. Linear logic. *Theoretical Computer Science* 50:1–102.

[Groenendijk and Stokhof, 1990] Groenendijk, J. and Stokhof, M. 1990. Dynamic montague grammar. In Kalman, L. and Polos, L., editors 1990, *Papers from the Second Symposium on Logic and Language.* Akademiai Kiadoo, Budapest. 3–48.

[Groenendijk and Stokhof, 1991] Groenendijk, J. and Stokhof, M. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14:39–100.

[Grosz *et al.*, 1983] Grosz, B. J.; Joshi, A. K.; and Weinstein, S. 1983. Providing a unified account of definite np anaphora. In *Proc 21st ACL.* Assoc for Comp Ling.

[Guenthner and Lehmann, 1983] Guenthner, F. and Lehmann, H. 1983. Rules for pronominalisation. In *EACL 83, Pisa, Italy.* 144–151.

[Guenthner and Lehmann, 1984] Guenthner, F. and Lehmann, H. 1984. Automatic construction of discourse representation structures. In *COLING 84, Stanford.* 398–401.

[Haegeman, 1991] Haegeman, L. 1991. *An Introduction to Government and Binding Theory.* Basil Blackwell, first edition.

[Hale and Keyser, 1986] Hale, K. and Keyser, J. 1986. Some transitivity alternations in english. Lexicon Project Working Papers 7, MIT, Center for Cognitive Science.

[Hale and Keyser, 1987] Hale, K. and Keyser, J. 1987. A view from the middle. Lexicon Project Working Papers 10, MIT, Center for Cognitive Science.

[Halvorsen, 1983] Halvorsen, P.K. 1983. Semantics for lfg. *Linguistic Inquiry* 14:567–615.

[Hardt, 1992] Hardt, D. 1992. An algorithm for vp ellipsis. In *Proceedings 30th ACL, Newark.* 9–14.

[Heim, 1982] Heim, I. 1982. *The Semantics of Definite and Indefinite Noun Phrases.* Ph.D. Dissertation, University of Massachusetts, Amherst.

[Herzog and Rollinger, 1991] Herzog, O. and Rollinger, C.-R., editors 1991. *Text Understanding in LILOG.* Springer-Verlag, Berlin.

[Hindle and Rooth, 1983] Hindle, D. and Rooth, M 1983. Structural ambiguity and lexical relations. *Computational Linguistics, 19* 103–120.

[Hobbs *et al.*,] Hobbs, Jerry; Stickel, Mark; Martin, Paul; and Edwards, Douglas up . Interpretation as Abduction. In *Proceedings, 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York. 95–103.

[Hobbs, 1991] Hobbs, Jerry 1991. Description of the Tacitus system as used for MUC-3. In *Proceedings, Third Message Understanding Conference, May 1991.* Morgan Kaufmann, Inc., San Mateo, California. 200–206.

[Hwang, 1992] Hwang, C.H. 1992. *A Logical Approach to Narrative Understanding.* Ph.D. Dissertation, University of Alberta.

[Jaspars, 1994] Jaspars, J.O.M. 1994. *Calculi for Constructive Communication.* Ph.D. Dissertation, ITK, Tilburg and ILLC, Amsterdam.

[Johnson and Kay, 1990] Johnson, M. and Kay, M. 1990. Semantic abstraction and anaphora. In *Proceedings of the 13th Meeting of the ACL*, Helsinki.

[Johnson and Klein, 1986] Johnson, M. and Klein, E. 1986. Discourse, anaphora and parsing. In *COLING 86, Proceedings of the Conference, Bonn, Germany.* 669–675.

[Kameyama, July 1994] Kameyama, M. 1994. Indefeasible semantics and defeasible pragmatics. Technical Report CS–R9441, CWI, Amsterdam.

[Kamp and Partee, 1995] Kamp, H. and Partee, B. 1995. Prototype theory and compositionality. to appear in Cognition 1995.

[Kamp and Reyle, 1991] Kamp, H. and Reyle, U. 1991. A calculus for first order discourse representation structures. Arbeitspapiere des Sonderforschungsbereichs 340 16, IMS Stuttgart, Germany.

[Kamp and Reyle, 1993] Kamp, H. and Reyle, U. 1993. *From Discourse to Logic.* Kluwer, Dordrecht.

[Kamp and Roßdeutscher, 1992] Kamp, H. and Roßdeutscher, A. 1992. Remarks on lexical structure,drs-construction and lexically driven inferences. Arbeitspapiere des Sonderforschungsbereichs 340 21, IMS Stuttgart, Germany.

[Kamp and Roßdeutscher, 1994a] Kamp, H. and Roßdeutscher, A. 1994a. Drs-construction and lexically driven inferences. to appear in: (ed) H. Schnelle, Lexical Semantics, special issue of Theoretical Linguistics.

[Kamp and Roßdeutscher, 1994b] Kamp, H. and Roßdeutscher, A. 1994b. Remarks on lexical structure and drs-construction. to appear in: (ed) H. Schnelle, Lexical Semantics, special issue of Theoretical Linguistics.

[Kamp, 1975] Kamp, H. 1975. Two theories about adjectives. In Keenan, E., editor 1975, *Formal Semantics for Natural Language.* Cambridge University Press, Cambridge. 123–155.

[Kamp, 1981a] Kamp, H. 1981a. The paradox of the heap. In Mönnich, U., editor 1981a, *Aspects of Philosophical Logic.* D. Reidel. 225–277.

[Kamp, 1981b] Kamp, H. 1981b. A theory of truth and semantic representation. In Groenendijk, J. and others, , editors 1981b, *Formal Methods in the Study of Language.* Mathematisch Centrum, Amsterdam.

[Kamp, 1992] Kamp, H. 1992. Disambiguation in discourse. In Aurnague, M.; Borillo, A.; Borillo, M.; and Bras, M., editors 1992, *Semantics of Time, Space and Movement. Working Papers of the 4th International Workshop.* Groupe "Langue, Raisonment, Calcul", CNRS et Université Paul Sabatier, CNRS et Université Toulouse Le Mirail. 281–308.

[Kaplan and Bresnan, 1982] Kaplan, R.M. and Bresnan, J. 1982. Lexical functional grammar. In Bresnan, J., editor 1982, *The mental representation of grammatical relations.* MIT Press, Cambridge Mass. 173–281.

[Kaplan, 1977] Kaplan, D. 1977. Demonstratives. an essay on the semantics, logic, metaphysics and epistemology of demonstratives and other indexicals. Unpublished manuscript, University of California, Los Angeles.

[Kasper, 1991] Kasper, W. 1991. The construction of semantic representations from f-structures. In Bes, G. G. and Guiillotin, T., editors 1991, *A Natural Language and Graphics Interface.* Springer-Verlag, Berlin. 122–146.

[Keller, 1988] Keller, W.R. 1988. Nested cooper storage. In Reyle, U. and Rohrer, C., editors 1988, *Natural Language Parsing and Linguistic Theory.* D. Reidel, Dordrecht, The Netherlands. 432–447.

[König, 1994a] König, E. 1994a. A study in grammar design. Arbeitspapiere des Sonderforschungsbereichs 340 54, IMS Stuttgart, Germany.

[König, 1994b] König, E. 1994b. Syntactic-head-driven generation. In *COLING 94, Kyoto, Japan.* 475–481.

[Krifka, 1994] Krifka, M. 1994. Parameterized sum individuals for plural reference and partitive quantification. University of Texas at Austin and IBM Germany Scientific Centre, Heidelberg.

[Kripke, 1972] Kripke, S.A. 1972. Naming and necessity. In Davidson, D. and Harman, G., editors 1972, *Semantics of Natural Language.* Reidel, Dordrecht. 253–355.

[Lang, 1985] Lang, E. 1985. Symmetrische prädikate: Lexikoneintrag und interpretationsspielraum. eine fallstudie zur semantik der personenstandslexik. *Linguistische Studien, A 127* 75–113.

[Lascarides and Asher, 1993] Lascarides, A. and Asher, N. 1993. A semantics and pragmatics for the pluperfect. In *EACL 93*, Utrecht. 250–259.

[Lascarides and Oberlander, 1993] Lascarides, A. and Oberlander, J. 1993. Temporal connectives in a discourse context. In *EACL 93*, Utrecht. 260–268.

[Laubsch and Nerbonne, 1991] Laubsch, J. and Nerbonne, J. 1991. An overview of $\mathcal{NLL}$. Technical report, Hewlett-Packard Laboratories.

[Lehmann, 1988] Lehmann, H. 1988. The lex-project - concepts and results. In Blaser, A., editor 1988, *Natural Language at the Computer.* Springer-Verlag, Berlin. 122–146.

[Levine, 1993] Levine, B. 1993. *English Verb Classes and Alternations: A Preliminary Investigation.* The University of Chicago, Chicago.

[Lowden *et al.*, 1993a] Lowden, B.G.T.; Walls, B.R.; De Roeck, A.N.; and Fox, C.J. 1993a. An approach to paraphrasing logical query languages in English. *Journal of Database Technology* 4(4):227–233. Dated as 1991/1992.

[Lowden *et al.*, 1993b] Lowden, B.G.T.; Walls, B.R.; De Roeck, A.N.; and Fox, C.J. 1993b. Modal reasoning in relational systems. *Journal of Database Technology* 4(4):235–244. Dated as 1991/1992.

[Martin-Löf, 1984] Martin-Löf, P. 1984. *Intuitionistic Type Theory.* Bibliopolis.

[May, 1985] May, R. 1985. *Logical Form.* MIT Press, Cambridge Mass.

[McAllester and Givan, 1992] McAllester, D. and Givan, R. 1992. Natural language syntax and first order inference. *Artificial Intelligence* 56:1–20.

[McKeown *et al.*, 1994] McKeown, K.; Kukich, K.; and Shaw, J. 1994. Practical issues in automatic documentation generation. In *Proceedings 4th Conference on Applied NLP*, ACL, Stuttgart, Germany. 7–14.

[Mellish, 1985] Mellish, C. 1985. *Computer interpretation of natural language descriptions.* Ellis Horwood, Chichester, Sussex.

[Mellish, 1988] Mellish, C. 1988. Implementing systemic classification by unification. *Computational Linguistics* 14:40–51.

[Millies and Pinkal, 1993] Millies, S. and Pinkal, M. 1993. Linking one semantic interpretation system to different syntactic formalisms. Extended Abstract of the Dagstuhl Colloquium on Semantic Formalisms in Natural Language Processing.

[Montague, 1963] Montague, R. 1963. Syntactic treatments of modality, with corollaries of reflexion principles and finite axiomatizability. *Acta Philosophica Fennica* 16:153–167.

[Montague, 1970] Montague, R. 1970. Universal grammar. *Theoria* 36:373–398.

[Montague, 1973] Montague, R. 1973. The proper treatment of quantification in ordinary english. In e.a., J. Hintikka, editor 1973, *Approaches to Natural Language.* Reidel. 221–242.

[Moore, 1989] Moore, R. C. 1989. Unification-based semantic interpretation. In *Proceedings 27th Annual Meeting of the Association for Computational Linguistics.* 33–41.

[Muskens, 1989] Muskens, R. 1989. *Meaning and Partiality.* Ph.D. Dissertation, University of Amsterdam.

[Muskens, 1994] Muskens, R. 1994. A compositional discourse representation theory. In Dekker, P. and Stokhof, M., editors 1994, *Proceedings 9th Amsterdam Colloquium.* ILLC, Amsterdam. 467–486.

[Nerbonne, 1991] Nerbonne, J. 1991. Constraint-based semantics. In *Proceedings 8th Amsterdam Colloquium.* 425–443.

[Nerbonne, 1992] Nerbonne, J. 1992. $\mathcal{NLL}$ models. Technical report, DFKI Saarbrücken.

[Pereira and Shieber, 1987] Pereira, F.C.N. and Shieber, S.M. 1987. *Prolog and Natural Language Analysis*, volume 10 of *CSLI Lecture Notes.* CSLI, Stanford. Distributed by University of Chicago Press.

[Pinkal, 1985] Pinkal, M. 1985. *Logik und Lexikon: Die Semantik des Unbestimmten.* de Gruyter, Berlin.

[Pinkal, 1991a] Pinkal, M. 1991a. On the syntactic-semantic analysis of bound anaphora. In *Proceedings of the 5th Meeting of the European ACL (EACL)*, Berlin.

[Pinkal, 1991b] Pinkal, M. 1991b. Vagheit und Ambiguität. In Stechow, A.von and Wunderlich, D., editors 1991b, *Semantik/Semantics.* de Gruyter, Berlin and New York. chapter 11, 250–269.

[Poesio, 1994] Poesio, M. 1994. *Discourse Interpretation and the Scope of Operators.* Ph.D. Dissertation, University of Rochester, Department of Computer Science.

[Pollard and Sag, 1994] Pollard, C. and Sag, I. 1994. *Head-Driven Phrase Structure Grammar.* CSLI Lecture Notes. CSLI, Stanford. Distributed by University of Chicago Press.

[Pustejowsky, 1991] Pustejowsky, J. 1991. The generative lexicon. *Computational Linguistics* 17(4):409–441.

[Rappaport and Levine, 1986] Rappaport, M. and Levine, B. 1986. What to do with theta-roles. Lexicon Project Working Papers 11, MIT, Center for Cognitive Science.

[Reyle and Gabbay, 1994] Reyle, U. and Gabbay, D. 1994. Direct deductive computation on discourse represenation structures. *Linguistics and Philosophy* 17:343–390.

[Reyle, 1985] Reyle, U. 1985. Grammatical functions, discourse referents and quantification. In *IJCAI 85.* 829–831.

[Reyle, 1993a] Reyle, U. 1993a. Dealing with ambiguities by underspecification: Construction, representation and deduction. *Journal of Semantics* 10:123–179.

[Reyle, 1993b] Reyle, U. 1993b. Vagueness and ambiguity in drt. *Journal of Semantics.*

[Rooth, 1987] Rooth, M. 1987. Np interpretation in montague grammar, file change semantics and situation semantics. In Gärdenfors, P., editor 1987, *Generalized Quantifiers: linguistic and logical approaches.* Reidel, Dordrecht. 237–269.

[Rosetta, 1994] Rosetta, M.T. 1994. *Compositional Translation.* Kluwer, Dordrecht.

[Roßdeutscher, 1994] Roßdeutscher, A. 1994. Fat child meets drt. a semantic representation for the opening lines of kaschnitz' "das dicke kind". to appear in: (ed) H. Schnelle, Lexical Semantics, special issue of Theoretical Linguistics.

[Sæbø, 1993] Sæbø, K.J. 1993. Anaphoric presuppositions, accommodation, and zero anaphora. In Kamp, H., editor 1993, *DYANA-2 Deliverable R2.2A, Part II.* The DYANA-2 Project Administrator, ILLC, University of Amsterdam. 87–94.

[Sandt, 1992] Sandt, R.A. van der 1992. Presupposition projection as anaphora resolution. *Journal of Semantics* 9:333–377. Special Issue: Presupposition, Part 2.

[Saurer, 1993] Saurer, W. 1993. A natural deduction system of discourse representation theory. *Journal of Philosophical Logic* 22(3):249–302.

[Schank, 1972] Schank, R. 1972. Conceptual dependency: a theory of natural language understanding. *Cognitive Psychology* 3:552–631.

[Schielen, 1993] Schielen, M. 1993. Ein axiomatisierung der skopusbestimmung zur constraintbasierten erzeugung der logischen form. Master's thesis, IMS, University of Stuttgart.

[Schubert and Pelletier, 1982] Schubert, L. K. and Pelletier, F. J. 1982. From english to logic: Context-free computation of 'conventional' logical translation. *American Journal of Computational Linguistics* 8:26–44.

[Sedogbo, 1988] Sedogbo, C. 1988. Sylog: A drt system in prolog. In Dahl, V. and Saint-Dizier, P., editors 1988, *Natural Language Understanding and Logical Programming II.* Elsevier Science Publishers. 185–201.

[Seidenberg *et al.*, 1982] Seidenberg, M. S.; Tanenhaus, M. K.; Leiman, J.; and Bienkowski, M. 1982. Automatic access of the meanings of ambiguous words in context: some limitations of knowledge-based processing. *Cognitive Psychology* 14:489–537.

[Shoham, 1988] Shoham, Y. 1988. *Reasoning about Change.* MIT Press, Cambridge, Mass.

[Sidner, 1983] Sidner, C. 1983. Focusing in the comprehension of definite anaphora. In Berwick, R. and Brady, M., editors 1983, *Computational Models of Discourse.* MIT Press.

[Smith, 1990] Smith, W. 1990. Handling constrained clauses in discourse representation theory. Artificial Intelligence Research Report AI-1990-02, The University of Georgia, Athens, Georgia.

[Steel and De Roeck, 1987] Steel, Sam and De Roeck, Anne N. 1987. Bidirectional chart parsing. In Mellish, Christopher S. and Hallam, John, editors 1987, *Advances in Artificial Intelligence (Proceedings of AISB87)*, Chichester. Wiley. 223–235.

[Swinney, 1979] Swinney, D. A. 1979. Lexical access during sentence comprehension: (re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior* 18:545–567.

[Tarski, 1956] Tarski, A. 1956. The concept of truth in the languages of the deductive sciences. In Woodger, J., editor 1956, *Logic, Semantics, Metamathematics.* Oxford. First published in Polish in 1933.

[Thomason, 1980] Thomason, R. 1980. A note on syntactical treatments of modality. *Synthese* 44:391–395.

[Turner, 1988] Turner, R. 1988. A theory of properties. *The Journal of Symbolic Logic* 54.

[Turner, 1992] Turner, R. 1992. Properties, propositions and semantic theory. In Rosner, M. and Johnson, R., editors 1992, *Computational Linguistics and Formal Semantics.* CUP, Cambridge.

[Wada and Asher, 1986] Wada, H. and Asher, N. 1986. Buildrs: An implementation of dr theory and lfg. In *COLING 86.* 540–545.

[Wada, 1990] Wada, H. 1990. Discourse processing in mt. In *COLING 90.* 73–75.

[Woods, 1977] Woods, W. 1977. Semantics and quantification in natural language question answering. In *Advances in Computers. Volume 17.* Academic Press. 1–87.

[Wunderlich, 1987] Wunderlich, D. 1987. An investigation of lexical composition: The case of german be-verbs. *Linguistics* 25:283–331.

[Zeevat *et al.*, 1987] Zeevat, H.; Klein, E.; and Calder, J. 1987. Unification categorial grammar. In Haddock, N.; Klein, E.; and Morrill, G., editors 1987, *Categorial Grammar, Unification Grammar and Parsing.* Centre for Cognitive Science, University of Edinburgh. 195–222.

[Zeevat, 1989] Zeevat, H. 1989. A compositional approach to discourse representation theory. *Linguistics and Philosophy* 12:95–131.

[Zeevat, 1992] Zeevat, H. 1992. Presupposition and accommodation in update semantics. *Journal of Semantics* 9(4):379–412. Special Issue: Presupposition, Part 2.