Syntactic and Semantic Underspecification in the Verb Phrase

Lutz Marten

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

University of London

June 1999

Department of Linguistics
School of Oriental and African Studies

Abstract

This thesis is concerned with verbs and the relation between verbs and their complements. Syntactic evidence is presented which shows that the distinction between arguments and adjuncts reflects the optionality of adjuncts, but that adjuncts, once introduced, behave as arguments of the verb. An analysis is proposed which reflects this observation by assuming that verbal subcategorization is underspecified, so that optional constituents can be introduced into the verb phrase. The analysis is developed within a formal model of utterance interpretation, Labelled Deductive Systems for Natural Language (LDSNL), proposed in Kempson, Meyer-Viol & Gabbay (1999), which models the structural aspect of utterance interpretation as a dynamic process of tree growth during which lexical information is combined into more complex structures which provide vehicles for interpretation, propositional forms. The contribution of this thesis from the perspective of utterance interpretation is that it explores the notion of structural underspecification with respect to predicate–argument structure. After providing a formalization of underspecified verbal subcategorization, the thesis explores the consequences this analysis of verbs and verb phrases has for the process of tree growth, and how underspecified verbs are interpreted. The main argument developed is that verbs syntactically encode the possibilty for pragmatic enrichment; verbs address mental concepts only indirectly, so that the establishment of their eventual meaning, and, therefore, their eventual arity is mediated by the cognitive process of concept formation. Additional support for this view is provided by an analysis of applied verbs in Swahili which, from the perspective adopted here, can be seen to encode an explicit instruction for concept strengthening, an instruction to the hearer to derive additional inferential effects. The analysis presented in this thesis thus supports the view that natural language interpretation is a process in which structural properties and inferential activity are thoroughly intertwined.

Acknowlegdments

This thesis would not exist without the encouragement, support, and inspiration I have received from a number of directions over the years.

For financial and institutional support I am indebted to the University of London Postgraduate Trust and the British Academy for offering me studentships here in London. I furthermore gratefully acknowledge financial support for travelling to East Africa from the DAAD (1995) and from the Irwin Trust (Central Research Fund, University of London), SOAS, and the Department of Linguistics at SOAS (1997).

Intellectually, academically, personally, I have benefitted first and foremost from the supervision of my work by Ruth Kempson, whose insight, vision and broadness of perspective constituted a major reason for my coming to, and still being in, London. The Department of Linguistics provides the most stimulating and cooperative atmosphere for pursuing academic studies; I wish to thank especially Monik Charette and Thea Bynon, as well as Dave Bennett, Wynn Chao, Dick Hayward, Bruce Ingham, Jonathan Kaye, and Shalom Lappin for making my stay here so pleasurable.

In addition to intellectual stimulation, I have found a friend; David Swinburne has been a source of advice, consolation and, when needed, accomodation, which, Inshallah, will continue to flow. My life and times at SOAS would furthermore not have been the same without Trevor Marchand, Taeko Maeda, Stefan Ploch, Denise Perrett, Wilfried Meyer-Viol, Tony Hunter, Andrew Simpson, and Lucy Walters.

The thesis results partly from my interest in the Swahili language. I wish to thank everybody who shared this interest with me and helped me to better understand both language and culture, in particular Sauda Barwani, Ridder Samsom, Abdulla Othman Ahmed and his family, Lewis Lukindo, Haroub Nassor Hamoud, Donavan McGrath, as well as the Taasisi ya Kiswahili, Zanzibar.

Above all I wish to thank my family – my parents for their love, understanding, support and unwavering commitment to my well-being, my future wife, I hope, Nancy, for her love, her cheerfulness, her energy, and her brightness for me to share, and my grandmother, Erna Dalkowski, to whom I hereby dedicate this thesis.

Contents

Chapter 1

Introduction

1. Introduction

This thesis is concerned with aspects of the syntax, semantics, and pragmatics of the verb phrase, as seen, in particular, from the perspective of the formal model of utterance interpretation Labelled Deductive Systems for Natural Language (LDSNL), developed in Kempson, Meyer-Viol & Gabbay (1999). The central question investigated is how nominal constituents of the verb phrase, taken here to include noun phrases and prepositional phrases, are licensed and interpreted, and how they interact with information provided by the verb. The discussion thus turns around the question of verbal subcategorization, optional and obligatory constituents, and the function of prepositions, as illustrated in the following examples:

(1a) Fran was baking a cake for Mary in the oven.
(1b) Sally put the flowers on the table with a vengeance.
(1c) The McDonalds live in a house by the seaside.

What all these examples have in common is that the verb is combined with more constituents than seem to be required. The sentences also show that although PPs are most often optional, some PPs are required by the verb's subcategorization information, e.g. *on the table* in (1b). The analysis of sentences like those in (1) is the main topic of this work.

The theoretical perspective adopted is that of LDSNL, a formal model of utterance interpretation which provides an explicit characterization of the process by which hearers access natural language words in the order in which they appear in the utterance and use the information provided to build structured semantic representations in a step-by-step fashion. This dynamic perspective on structure building places certain restrictions on the analysis of how verbs and their complements combine, since the process has to be modelled as proceeding strictly incrementally. Furthermore, the process is goal-driven, guided by the overall requirement that hearers establish propositional structures to derive inferential effects from the words encountered. This also implies that tree structures interact directly with pragmatic reasoning. The approach developed in this thesis is that the pragmatic process of enrichment, which enables hearers to construct occasion specific conceptual representations, plays a central role in the interpretation of natural language verbs and verb phrases. The overall achievement of the thesis is thus that it provides a unified analysis of verb phrase adjunction for LDSNL which integrates the syntactic, semantic, and pragmatic aspects of the interpretation process.

In the next two sections, I introduce the LDSNL model in more detail. The final section provides an overview of the thesis.

2. LDSNL: Conceptual Assumptions

In this section I discuss the conceptual assumptions underlying the LDSNL model. One of the central assumptions of LDSNL is that knowledge of language can at least partly be characterized as the ability to assign mental representations of meaning to incoming lexical information in the process of utterance interpretation. I discuss briefly how this view relates to the notions of competence and performance. I then introduce the Government Phonology view that the cognitive role of phonology is to assign phonological structure to incoming physical signals, and to provide access to lexical information (Kaye 1989). After this discussion I provide an introduction to Relevance Theory (Sperber & Wilson 1986/1995), which characterizes inferential aspects of communication as resulting from general cognitive constraints on information processing. I discuss how the structure building process modelled in LDSNL relates to the Relevance theoretic conception of the role of the hearer in the establishment of representational structure. I finally provide a brief summary of the overall model of utterance interpretation which results from the preceding discussion.

2.1. A Model of Utterance Interpretation

The main concern of LDSNL is to model the syntactic aspects of the process of utterance interpretation. In the broadest sense, utterance interpretation involves an incoming signal, prototypically a continuous undivided input stream of sound, on the one end, and a completely interpretable enriched mental representation on the other end. A preliminary sketch might look as (2):

(2) *Utterance Interpretation (first version)*

sound -> (phonology, syntax, semantics, pragmatics) -> interpretation

The sketch in (2) shows that the mapping from sound to meaning involves as intermediate steps the application of phonological, syntactic, semantic, and pragmatic knowledge, in that all of them contribute to the processing of some input. However, from the perspective of utterance interpretation, the interesting question with respect to this knowledge is not so much the independent characterization of each kind of knowledge, but the contribution to deriving an interpretation for the signal; a dynamic perspective highlights the relationship between different kinds of linguistic knowledge in fulfilling an overall task. If, and to what extent, different kinds of knowledge can be characterized as being distinct components or modules can then be characterized with reference to their particular contribution to the building of interpretations.

Before looking further at the components, I discuss in the next section the underlying claim in the sketch in (2) that it is possible to study linguistic knowledge from the point of view of the hearer, without looking at production, or competence.

2.2. Competence and Performance

The study of utterance interpretation concerns to some extent performance, since it is concerned with how language is used and thus puts language into a functional perspective. On the other hand, the question of what enables hearers to perform the task of deriving interpretations is a question about knowledge, or competence. Competence and performance are concepts most closely associated with Chomsky (1957, 1964, and elsewhere). Chomsky's original idea is embedded in a more general conception of linguistics, involving the two idealizations that language is used in a homogeneous speech community, and that there is an ideal speaker–hearer relation. Against this background, competence is characterized by the ability to produce and understand a potentially infinite number of novel sentences, and is contrasted with performance, which involves competence plus a number of non-linguistic factors such as limitations of memory, distortion in the speech channel, bad sentence planning and others. In order to find out about linguistic knowledge on which speakers draw, the linguist has to abstract away from performance factors, and postulates a body of mental principles or rules which determine the set of all possible ('well-formed') sentences. The well-formedness of sentences is checked against grammaticality judgements of speakers. In addition, Chomsky assumes that syntactic knowledge is encapsulated, that is to say, it constitutes a distinct mental module which operates independently of other modules. In recent writings, Chomsky (1995) points out that the analysis of syntactic competence should only postulate components which are 'virtually conceptually necessary', namely 1) an interface with the auditory–perceptual system, 2) an interface with the conceptual–intentional level, and 3) an interface with the lexicon.

Competence in the Chomskian sense, then, is the innate, abstract knowledge, e.g. a body of principles and fixed parameters, of speakers of a given language, which enables them to understand and produce an infinite number of novel sentences. This knowledge is autonomous, that is, is not determined by, or similar to, any other mental facilities, but its design is constrained by the virtual conceptual necessities of three interfaces; sound, meaning, and the lexicon. This conception gives rise to the 'T-model', which has in its basic form (that is, with or without 'deep' and 'surface' structure) been assumed at all phases in generative linguistics. The interface to the conceptual–intentional system is the level of logical form (LF), the interface to the auditory–perceptual module is the level of phonetic form (PF) and the interface to the lexicon is at the bottom, without its own level:

(3) *T-model*

```
LF  PF
 |  /
 | /
 |
(lexicon)
```

The point where the paths to PF and LF branch corresponds traditionally to surface structure (S-structure), but in the more recent Minimalist Program to spell out, since no level of representation is assumed at this point.

Compared to the interpretation model (2) above, it seems that both PF and LF are part of interpretation. The interface levels could thus be just taken over and connected with, say, a line:

(4) *Utterance Interpretation (rejected version)*

sound -> phonology -> PF – LF -> (semantics, pragmatics) -> interpretation

In (4) 'syntax' is replaced with 'PF – LF'. However, the immediate problem is that there would be no words in the interpretations derived – there is no interface to the lexicon[1]. This is not a mere technical problem. Rather, it follows from the competence–performance distinction: the T-model represents competence only, it is not intended to be related, or even relatable to language use[2]. The relation between competence and performance is achieved by different, additional knowledge, for example parsers. A parser might make use of competence, but functions independently of it. However, there is no need under the Chomskian conception to have the (theoretical) characterization of knowledge of language be influenced by (psycholinguistic) evidence of language use[3]. There is also no need to incorporate the model of grammar into 'the model of the mind'; that is, since grammar, and syntax in particular, is encapsulated, the relation of these systems to other cognitive systems (e.g. vision, general reasoning) is irrelevant, except for the rather weak characterization of the interface levels[4].

Of course, there is nothing wrong *per se* in assuming that humans have linguistic competence in the sense that we can classify sentences as right or wrong (as grammatical and ungrammatical), but it is not something which we do often, nor something which is a (functionally, evolutionarily, ...) sensible activity. It is, in this sense, not a virtual conceptual necessity for our cognitive make-up. On the other hand, we do use language to communicate, we act as speakers and hearers, and in order to do so, we employ knowledge. The shift in perspective advocated in LDSNL is to devise a theory which starts from the fact that in utterance interpretation, a physical structure (sound) is mapped onto a mental structure (a representation of meaning), so that the explanation can, or at least could, be measured against psycholinguistic data and is embedded, or at

1 There is also no room for 'movement', or purely syntactic derivations, that is from deep structure to surface structure, or from a 'numeration' to spell-out. Note, however, despite the importance of movement in most analyses within Chomskian linguistics, it is not, according to Chomsky (and presumably most non-Chomskian linguists) virtually conceptually necessary. That is, the 'PF – LF' notation might work even if most of Chomsky's assumptions are maintained, by shifting from derivation to representation, and from movement to chains (as proposed e.g. by Brody 1995). However, my reasons for rejecting the Chomskian conception are ultimately conceptual rather than technical.

2 Cf. e.g. Jackendoff (1998: 8).

3 See e.g. Jackendoff (1998) for discussion. Models of grammar which incoporate psycholinguistic evidence tend to depart from the respective classic generative model (e.g. Bresnan 1978, Berwick & Weinberg 1984, Gorrell 1995).

4 This is one of the main problems identified and discussed by Jackendoff (1998).

least potentially embeddable, into a larger theory of cognition. But from this perspective, utterance interpretation is not performance, at least not in the sense of limitations resulting from (lack of) concentration, or memory limitations. Rather, competence can be viewed, in contrast to the Chomskian conception, as the underlying ability of two distinct activities – speaking and hearing. Since these are two distinct activities, the respective underlying knowledge might in fact be different, although, of course, it would be somewhat surprising if it turned out to be two completely distinct systems of knowledge. Competence in the Chomskian sense can probably be reconstructed from the conception(s) of competence assumed here, but it is, cognitively, epiphenomenal. Throughout this thesis, I will thus assume that the knowledge modelled by linguistic theory is the knowledge which mediates between sound and meaning, in particular as used in building interpretations. In the next section, I turn more closely to the difference between interpretation and production, and try to show why it makes sense to restrict attention to interpretation.

2.3. Interpretation and Production

In the beginning of this section, I have introduced the LDSNL assumption that a theory of linguistic knowledge should start from the fact that language is used in communication, and that it involves, in utterance interpretation, a process, possibly involving several sub-systems, of mapping sound structures to interpretation. There are reasons for assuming that understanding is cognitively prior to production, and thus for focussing on utterance interpretation, rather than utterance production. First, there are the (pre-theoretic) considerations that in language acquisition, perception appears to precede production, and that in language impairment, perception is more robust than production. In language acquisition, children universally undergo a number of stages in production, using increasingly more complex structures. However, it seems plausible to assume that children are able to understand at any given stage utterances which are at least as complex as the ones they produce. This is also implied in most (all?) theories of language acquisition – whether one says that structures are bootstrapped from recurrent patterns, or that parameters are set from appropriate input, one is thereby committed to saying that children are able to parse some relevant input before it is acquired. In language impairment, on the other hand, there seems to be, in addition to the asymmetry between impairment of 'functional' and 'conceptual' systems (Broca's and Wernicke's areas), a less well-observed asymmetry between impairment of production and interpretation systems. And it appears that it is the latter, which is less often affected by impairment[5].

A second set of reasons for assuming the primacy of interpretation comes from theoretical work in phonology and in pragmatics.

2.4. Goverment Phonology

The role of phonology in cognitive linguistic theory in a way most compatible with the LDSNL view has been described by Kaye (1989, 1997). Kaye rejects the idea that phonology is based on articulatory phonetics, that is on an analysis of how humans produce speech sounds. In addition to phonological evidence, Kaye points out that the speech organs have no obvious parallel in our perceptual apparatus, so that a phonetics based model of phonology is essentially speaker based. As an alternative to a phonetics based model, Kaye proposes a model in which phonology is seen as a parsing device. Under this view, phonological knowledge serves to divide an incoming continuous input stream into phonological units which provide access to lexical entries. This is the view underlying Government Phonology, (GP) (cf. amongst others, Kaye 1989, 1995, Kaye, Lowenstamm & Vergnaud 1990, Charette 1991), so that the following discussion is largely based on work within this framework.

The view that the purpose of phonology is to parse an input stream as proposed in GP implies a particular view of morphology and the lexicon (Kaye 1995). Phonological information provided in the signal serves to identify phonological domains, which are representational units stored in the lexicon. Kaye (1995: 301-318), discussing the phonology–morphology interface, argues that there is in fact very little interplay between phonology and morphology, and that there are only two basic notions of morphological structure: analytic morphology presents the hearer with more than one phonological domain, while non-analytic morphology treats morphologically complex forms as only one phonological domain. An example of (one type of)[6] analytic morphology is provided by a compound like *blackboard* (5):

(5) [[black][board]]

The brackets in (5) serve to indicate domainhood, that is here, to indicate that all *black, board,* and *blackboard,* are treated as phonological domains. That

5 Pointed out to me by Marie-Ann Kemp in personal communication.

6 A second type of analytic morphology is discussed in Chapter 6.

means that the two parts of the compound, as well as the whole compound initiate a lexical search. Phonologically, domainhood in (5) follows from GP assumptions about the licensing and interpretation of phonological positions, which serve to provide the hearer with parsing cues as to the lexical search requirement when hearing compounds like *blackboard*.

An example of non-analytic morphology is given in (6):

(6) [parental]

The claim here is that there is only one phonological domain for *parental*. The form cannot be computed and has to be looked up in the lexicon. It is identical in this respect to words like *agenda* or *advantage*. This, of course, leaves the question about the relation between a word like *parental* and its 'source' *parent*. The answer Kaye provides has to do with the organization of the lexicon. Following Kaye & Vergnaud (1990), Kaye (1995: 321) proposes:

> ... phonological representations do not form part of lexical representations as such but are rather the addressing system for lexical access. A phonological representation is the address of a lexical entry. [...] Lexical items that are phonologically similar are physically proximate in the psychological lexicon.

That is, forms like *parent* and *parental*, as well as irregular morphology like *keep – kept*, are not phonologically derived from one underlying representation. They constitute different, simplex phonological domains and hence address separate lexical entries. However, since the lexicon is (partly) organized according to phonological similarity, a provision such as 'proximate lexical entries are easily accessible' ensures that the information of both *parental* and *parent* is accessed at low cost.

Since phonology in this view accesses lexical information, the different types of morphology can be seen as different routes to the lexicon. Non-analytic morphology is an instruction to address the lexical entry directly, and morphological complexity, if any, has to be stored under this entry. Analytic morphology allows the hearer to compute the meaning contributed by several domains.

As has been argued in GP, many phonological processes can be viewed as providing the hearer with parsing clues about lexical access, for example domain final empty nuclei in English, or final obstruent devoicing in languages such as German or Russian.

For a model of utterance interpretation as developed here, the GP view of phonology offers two interesting points. First, to characterize phonology as a tool for parsing and lexical access entails, in line with the argument presented in this section, that interpretation is cognitively prior to utterance production. The nature of phonology is such that it enables hearers to decode information, so that a speaker, in order to encode, has to be able to decode.

Secondly, this view is highly compatible with the interpretation analysis of linguistic knowledge. The view of phonology developed in GP can be expressed as in the following sketch:

(7) *Utterance Interpretation (second version)*

sound --> phonology --> lexicon (syntax, semantics, pragmatics) --> interpretation

That is, phonology is a mapping from incoming sound to the lexicon; phonological knowledge under this view provides access to lexical information. As can be seen from (7), the view overcomes the problem with the T-model noted above, since it is the interface to the lexicon which makes the T-model difficult to be integrated into the model of utterance interpretation proposed in LDSNL. In (7), the lexical entries identified by phonological domains can be seen as input to further processing. It is this process of structure building from the lexicon which is modelled in LDSNL. Before turning to LDSNL, however, I discuss how the primacy of interpretation (as opposed to production) is motivated from Relevance Theory.

2.5. Relevance Theory

The LDSNL model is closely linked to Relevance Theory (RT), (Sperber & Wilson 1986/1995), in that it provides a model of syntactic knowledge based on the Relevance theoretic assumption that utterance interpretation is a goal directed process. I will make extensive reference to work in RT below, particularly for the discussion of concepts and concept formation in Chapter 5. In the section here, I am mainly concerned with the overall cognitive model proposed in RT, and how it relates to the LDSNL perspective.

Relevance Theory is a cognitive theory, where pragmatic aspects of natural language interpretation are explained by principles of cognition. RT takes the work of Grice (1967, 1989) as its historic antecedent. In particular, Sperber & Wilson (1986/95) argue, following Grice, that commmunication involves inference on the part of the hearer, who has to work out the speaker's intended meaning. Simple decoding, on its own, is not enough to recover meaning. However, RT differs from Grice's conception in important respects. I outline Grice's position first, and then discuss how RT departs from it.

Grice proposes that some aspects of communication involve inference on the part of the hearer, so that, in addition to decoding the meaning of sentences, hearers derive implicatures in interpretation to establish the full meaning of an utterance. The inferential aspects of interpretation follow, according to Grice, from the assumption that certain conversational rules are being obeyed by speakers and hearers. In particular Grice proposes that communication is governed by a co-operative principle, which instructs speakers as follows: "Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged" (Grice 1989: 26). The principle can be further specified by a number of rules, grouped under four 'maxims'. Grice proposes the following rules (Grice 1989, quoted from Sperber & Wilson 1995: 33/34):

(8) *Grice's Maxims of Conversation*

Maxims of quantity
1. Make your contribution as informative as is required (for the current purpose of the exchange).
2. Do not make your contribution more informative than is required.

Maxims of quality
Supermaxim: Try to make your contribution one that is true.
1. Do not say what you believe to be false.
2. Do not say that for which you lack adequate evidence.

Maxim of Relation
Be relevant.

Maxims of manner
Supermaxim: Be perspicuous.
1. Avoid obscurity of expression.
2. Avoid ambiguity.
3. Be brief (avoid unnecessary prolixity).
4. Be orderly.

While Sperber & Wilson agree with Grice that communication involves inference, they do not adopt the co-operative principle and maxims, for three reasons. First, it is not clear which status they have in linguistic or cognitive theory – are they learned or innate, universal or culture-specific, part of our linguistic or of our social knowledge? While the maxims of quality, for example, have an almost moral flavour, the maxims of manner sound rather more stylistic. Secondly, the maxims are comparatively vague. Thus, it is not clear how, for example, the maxims of manner can be made more precise. Furthermore, there seems to be a certain amount of overlap – the maxim of relation, 'be relevant', for example, probably involves some consideration of the quality in relation to the quantity of the utterance – but these aspects are expressed by different maxims. Lastly, and most importantly, Sperber & Wilson argue that inference not only plays a role in finding out what has been implied, but also in establishing what has been said in the first place, that is to say, inference is required even for the establishment of linguistic meaning, in addition to the establishment of inferences drawn from it. The role of non-demonstrative inferential reasoning in the establishment of what has been said, as opposed to what has been implied, includes cases of ambiguity resolution, reference assignment, where pronominal elements crucially underdetermine their encoded, truth-theoretic content, and the enrichment of encoded meaning, a process which will be discussed more extensively in Chapter 5.

The consideration of these questions leads Sperber & Wilson to propose a radically different view of pragmatics – they argue that inferential activities are all pervasive not only in communication, but also in the way we interact with our environment in general. The inferential abilities hearers use in establishing meaning in communication result, according to Sperber & Wilson, from the general cognitive abilities which are operative in information processing. Thus Sperber & Wilson propose that the inferential aspects of communication can be regarded as a reflex of principles of cognition. The argument is summarized below.

Humans are information processing animals. Input-modules (in the Fodorian (1981) sense) constantly extract information from the environment, largely automatically – we don't choose to see the things in front of us (unless we close our eyes), to smell a smell in the air, and we don't choose to process incoming natural language. This processing of incoming information results in a situation where at any given moment there is more sensory information than can be processed by central reasoning processes, where incoming information is projected. One of the central challenges for the human cognitive architecture is to make relatively fast and relatively reliable choices as to which incoming information is worth attending to, to distribute cognitive resources so as to improve our information state as efficiently as possible. In other words, we process maximally relevant information, our reasoning is goal directed (Sperber & Wilson 1986/95: 49):

> Our claim is that all human beings automatically aim at the most efficient information processing possible. This is so whether they are conscious of it or not; in fact, the very diverse and shifting conscious interests of individuals result from the pursuit of this permanent aim in changing conditions. In other words, an individual's particular cognitive goal at a given moment is always an instance of a more general goal: maximising the relevance of the information processed. [...]

With this observation in mind, Sperber & Wilson propose the Cognitive Principle of Relevance (Sperber & Wilson 1995: 260):

(9) *Cognitive Principle of Relevance*

Human cognition tends to be geared to the maximisation of relevance.

The relevance of a particular piece of information, where information can be characterized as a set of contextual assumptions, can be measured against the information state of the processor without these assumptions, i.e. before they are processed. If nothing changes, the gain in information is zero, hence processing the information is not relevant. On the other hand, if the new information changes the initial information state drastically, the information is very relevant. This change of information state can have a number of instantiations, depending on how exactly the new information interacts with old information – beliefs might be strengthened or contradicted, the new information might provide a premise to derive a conclusion which would not have followed from the initial information state. That is, relevance involves the maximization of contextual effects. But maximization on its own cannot explain how choices about which information to attend to can be made. Somehow or other, most information probably interacts with what we believe already in some way or other, so that it is inefficient to process all incoming information and check for potential contextual effects. Sperber & Wilson propose that maximization of contextual effects is counter balanced by processing cost. Mental activity involves 'cost' – thinking, information retrieval from long term memory, deriving conclusions are activities which need cognitive resources. These resources have to be allocated so as to derive maximally relevant information (in the maximal effect sense) with justified cognitive effort. This is expressed in the definition of relevance (Sperber & Wilson 1986/95: 125):

(10) *Relevance*

Extent Condition 1: an assumption is relevant in a context to the extent that its contextual effects in this context are large.
Extent Condition 2: an assumption is relevant in a context to the extent that the effort required to process it in this context is small.

The definition in (10) includes the two conditions on relevant information in a given context in two clauses: relevant information derives maximal contextual effects with minimal cognitive effort. The cognitive principle of relevance governs the relation between incoming data from the perceptual system (the input modules) and the central reasoning system. Note that the activity regulated by relevance is inferential, that is, contextual effects can be characterized as inferential potential of an assumption, cognitive effort is the cost associated with inferential activity. From this characterization of cognitive activity, Sperber & Wilson then develop a characterization of communication.

Communication involves cognitive activity. Sperber & Wilson's approach to characterize cognition as a basis for communication makes this relation more precise. In particular, since communication involves the processing of information, and since processing of information in general is geared towards maximization of relevance, as expressed in the cognitive principle of relevance, the very same principle can serve to explain the inferential-cognitive processes in communication. This approach answers those questions which were left open by Grice – what for Grice are a number of rather loose co-operative conventions is for Relevance theory cognitively mandatory. Our ability to handle communication (more or less successfully) results from our ability to handle information (more or less successfully). Both abilities result from our cognitive make up, not from social convention; both abilities are ultimately grounded in general reasoning, they are not part of linguistic knowledge or knowledge about language use.

One of Sperber & Wilson's basic assumptions about cognition is that there is always more information coming from the perceptual modules which could be processed than the amount of information which can actually be processed by the central reasoning system. Since incoming utterances are part of the incoming information, they compete with other data for the attention of the processor – the specialized, 'narrow' linguistic module is, after all, only another input module. However, there is a difference between just information and information communicated (or, more precisely, ostensively communicated); by addressing someone, we claim their attention. For the hearer this means that an ostensively communicated message (linguistic or otherwise) not only carries the content of that message, but also, and 'prior' to that content, it expresses the informative intention of the speaker. The hearer is justified to assume that the speaker, by addressing the hearer, implicitly claims that the content of the message will be relevant to the hearer (of course, it might turn out to be not as relevant to the hearer as the speaker had thought, but that is a different problem). This is expressed in the Communicative Principle of Relevance (Sperber & Wilson 1995: 260):

(11) *Communicative Principle of Relevance*

Every act of ostensive communication communicates the presumption of its own optimal relevance.

The phrase 'presumption of optimal relevance' is defined as follows (1995: 270):

(12) *Presumption of optimal relevance*

(a) The ostensive stimulus is relevant for it to be worth the addressee's effort to process it.

(b) The ostensive stimulus is the most relevant one compatible with the communicator's abilities and preferences.

That is, the hearer is justified to spend cognitive effort on processing a communicated message because she can assume that there are enough contextual effects to be derived to make the processing worthwhile. By the presumption of optimal relevance hearers can expect to derive maximally relevant inferential effects with no more than necessary cognitive cost since they can expect that the ostensive stimulus (i.e. in verbal communication, the utterance) used is the most relevant one possible in the given situation. The two principles of relevance thus highlight the relation between cognition and communication, since inferential reasoning in communication can be seen as a subcase of the more general cognitive constraint to process information efficiently.

Although the outline of relevance theory given so far is very brief, one important point for the present discussion can be noted, namely that in RT inferential abilities in communication are explained as resulting from cognitive abilities relevant for processing information, that is, from interpretation rather than from production. Sperber & Wilson derive communicative behaviour – as expressed in the communicative principle of relevance in (11) – from general cognitive behaviour, namely from our relevance-driven processing as embodied in the cognitive principle of relevance and definition of relevance. In other words, our ability to assess and choose information in linguistic communication is a reflex of our ability to handle information in general, but this latter ability does not presuppose ostensive stimuli – understanding is prior to informing.

There is another point which I would like to raise here, although the details will be more extensively discussed in Chapter 5, namely the relation between linguistic knowledge and pragmatics which is advocated in Relevance theory. In their formulation of relevance, Sperber & Wilson are very careful to retain the Gricean conception of the role of inference in utterance interpretation. The pragmatic aspects of utterance interpretation are inferential and involve the central reasoning system. However, other aspects of utterance interpretation are handled in the specialized linguistic module. These are automatic, algorithmic processes which crucially do not involve general reasoning, but the decoding of an arbitrarily defined code. The specialized linguistic module then provides input to the general cognitive system. Sperber & Wilson propose that the distinction between general reasoning and the linguistic system involves the distinction between non-demonstrative inference, the working mode of the general reasoning system, and decoding in the linguistic module. In view of the boundary between the two systems, Sperber & Wilson (1986/95: 185) argue that there are three aspects of utterance interpretation which require general reasoning, but which need to be resolved *before* a proposition can be established (where a proposition is a structure which can be evaluated for its truth value against a semantic model): disambiguation, reference assignment, and enrichment. That is, in contrast to Grice, Sperber & Wilson argue that non-demonstrative inference plays a role not only for recovering what has been implied by an utterance, but also to discover what has been said. The output of the linguistic module is a semantic representation, but "... semantic representations are incomplete logical forms, i.e. at best fragmentary representations of thought." (1986/95: 193). The first task of the central reasoning system is thus to derive a propositional form, to which (model-theoretic) content can be assigned, and only after that any implied meaning. On the other hand, the output of the linguistic system is not a proposition, but an underspecified logical form (LF), in need of disambiguation, reference assignment, and enrichment.

For utterance interpretation, this conception means that there is no full semantic representation for linguistic expressions without the contribution of pragmatic inferencing:

(13) *Utterance Interpretation (third version)*

sound -> phonology -> lexicon (syntax, pragmatics) -> {interpretation, semantics}

The sketch of information flow in (13) shows that the establishment of semantic representations is part of the interpretation process, to which all components before this process contribute.

2.6. Dynamic Syntax

Against this background, the LDSNL model is designed to provide an explicit characterization of the structure building processes required to use lexical information for the derivation of inferential effects. Hearers take information provided by lexical entries and use this information to build interpretations. This process is modelled as the incremental building of structured representations, reflecting the step-by-step contribution of lexical items to the establishment of the eventual representation. In accordance with Relevance theoretic assumptions about the nature of pragmatic inference, LDSNL structures do not represent a direct mapping from linguistic form to model theoretic interpretation. However, in contrast to Relevance theory, LDSNL does not employ a notion of interface level such as LF. Rather, the assumption is that pragmatic inferencing may apply to lexical items directly, as well as at each step of the process of structure building. This view implies that syntax and pragmatics derive propositional forms in tandem, so that pragmatic inference may determine the well-formedness of an LDSNL tree. These points are taken up in detail below in Chapter 5. For the moment, I assume that this assumption is correct[7].

The sketch of the information flow in utterance interpretation can thus be completed as given in (14):

---

7 As argued more extensively below, the motivation for a level of LF is more syntactic than pragmatic, so the real issue is to show that it is not possible, or at least not necessary to have it from the point of view of syntax. In contrast, the issue does not seem to be essential for Relevance Theory, so that in this chapter, where the emphasis is on providing a general picture of utterance interpretation, I do not discuss the issue in detail.

(14) *Utterance Interpretation (final version)*

sound -> phonology -> lexicon -> syntax/pragmatics -> {interpretation, semantics}

The final diagram in (14) is meant to describe the process of utterance interpretation as follows: Hearers receive a physical signal, a continuous input stream of sound, which provides the input to phonology. Phonology can be characterized as a body of knowledge which enables hearers to divide the input stream into phonological domains which provide lexical access. Lexical information provides the input to the building of the propositional form. The propositional form is established by using information from the lexicon and syntactically defined transition rules on the one hand, and non-demonstrative inference on the other. Model theoretic semantic interpretation is assigned to the propositional form, which is part of the interpretation of the utterance.

The syntactic aspect of utterance interpretation is modelled in LDSNL as an incremental increase of information about the eventual propositional form. The syntactic vehicle for interpretation are tree structures for which a (operational) semantics is given in the form of a modal logic, the logic of finite trees (LOFT). The growth of information in the process of utterance interpretation can be characterized as an increase in the information about the tree structure established at a given stage in the process. The formal tools of LDSNL introduced in the next section thus make reference to trees and tree descriptions, and characterize the increase of information about a given tree, corresponding to the process of tree growth. Transitions from one partial tree structure to another, up to the establishment of the eventual tree representing the propositional form, are licensed by lexically encoded instructions and by syntactically defined, optional transition rules. Before discussing the formal details of this process in the next section, I conclude this section by reviewing the basic conceptual assumptions underlying the LDSNL model.

The model of utterance interpretation discussed in this section reflects basic LDSNL assumptions about linguistic structure and knowledge of language, which are summarized here, serving as a conclusion to this section.

LDSNL shares with Relevance theory the commitment to a representational theory of mind. The tree descriptions with their corresponding tree structures built in LDSNL are structured representations of content, discrete from the natural language itself, and it is those representations over which the eventual semantic evaluation associated with the utterance are stated.

Furthermore, LDSNL places emphasis on the dynamic process of how the representation of linguistic structure is established. The process of structure building is defined as a goal-driven incremental process, during which the information provided from lexical items is used to build increasingly more articulated structures. In this sense, the building of tree structure is dynamic, so that syntax can be characterized as a set of transitions, rather than (or in addition to) a set of constraints on well-formed structures.

The model assigns a central role to the hearer in the process of utterance interpretation. This means that linguistic competence can be partly defined as the ability to assign structural representations to incoming linguistic information. This view is compatible with the conceptions proposed in Government Phonology and Relevance Theory and provides a conceptual link between knowledge of language and the function to which it is put.

In summary, LDSNL is a formal model of utterance interpretation in which linguistic competence is analysed as the ability to dynamically build structured representations of content. In the next section, I introduce the formal tools employed in LDSNL to articulate this view.

3. LDSNL: Formal Tools

In this section I introduce the formal tools employed in LDSNL and provide a sample derivation to show how the process of tree growth is modelled. I introduce the tree logic employed and the notion of declarative units, which annotate tree nodes. The dynamics of the system are characterized by the notions of task state and requirement, which are discussed before a set of transition rules and lexical actions are introduced, both of which drive the process of tree growth. A detailed sample derivation is given at that stage. I then introduce the analysis of dislocated constituents as underspecified tree location and the LINK operation, employed inter alia in the analysis of relative clauses.

3.1. Tree Logic

The dynamic unfolding of structure is modelled in LDSNL as tree growth, employing the logic of finite trees (LOFT) (Blackburn & Meyer-Viol 1994, Kempson et al. 1999). This is a modal logic which describes binary branching tree structures, reflecting the mode of semantic combination in function–application. Nodes in the tree may be identified by a numerical index ranging over 0 and 1:

(15)
```
        •0
       / \
   • 00    • 01
           / \
     • 010   • 011
```

By convention, the left daughter node of a node n is assigned the index n0 and the right daughter is assigned the index n1. Information holding at a given node may be described by a node description, or declarative unit (DU). The location of a node, i.e. its index, may be expressed by the predicate Tn (Treenode)[8]:

(16)
```
                 • {Tn(0), Q}
                     / \
   • {Tn(00), P}       • {Tn(01), P -> Q}
```

The DUs in (16) specify the tree location and information holding at that node. A left daughter is defined as an argument node, a right daughter as a functor node.

An alternative way to express the location of a DU is by using a subscript:

(17) $\{_{0}\, Q\}, \{_{00}\, P\}, \{_{01}\, P \text{->} Q\}$

The DUs in (17) describe the tree in (16).

The relation between tree nodes can be described by modal statements. This provides a means to state that some information holds at a daughter or at a mother node:

(18) $\{_{0}\, Q, \langle d_0 \rangle P, \langle d_1 \rangle P \text{->} Q\}$

The DU in (17) states that at Tn(0), Q holds, and that from the perspective of Tn(0), at the left daughter P holds, and at the right daughter P -> Q holds. The DU in (18) describes, again, the tree in (16). There are two basic modalities, one corresponding to the daughter relation (<d>, 'down'), and one corresponding to the mother relation (<u>, 'up'). These can be used with and without the numerical subscript, depending on whether it is important to distinguish

---

8 Note that in earlier versions of LDSNL the bullet ('•') was used to distinguish between facts and requirements in Declarative Units. In this thesis, the bullet is a graphic representation of a tree node; it is not part of the DU. Requirements are, as explained below, identified by a question mark, following more recent LDSNL usage.

between left and right branches. Furthermore, modality operators can be iterated, e.g. <d><d>, <d><u>, etc.

The system further allows for a weaker characterization of tree node relations, namely for saying that P holds somewhere down (or up), without specifying where exactly ('how deep down', or 'how high up') P holds. This is formally expressed by the 'Kleene star' operator, the reflexive transitive closure over the modalities <d> or <u>:

(19) $\langle d\rangle^{*}P =_{def} (P) \text{ or } \langle d\rangle\langle d\rangle^{*}(P)$

This recursive definition (and the analogous definition with the <u> operator) provides a means to express the underspecification of tree locations:

(20)
```
                  • {Tn(0), Q, <d>* R}
                          / \
        • {Tn(00), P}       • {Tn(01), P -> Q}

         {Tn(0*), R}
```

There are four DUs in (20), but only three of them are in fixed locations. The fourth DU is described as holding at Tn(0*) which is the numerical index indicating an unfixed daughter node of Tn(0). Correspondingly, the modal statement at Tn(0) indicates that at some unfixed daughter node R holds. This definition of underspecified tree location is the tool employed in LDSNL for the analysis of preposed constituents such as *wh*-pronouns or left dislocated topics. The analysis of verbal underspecification developed in this thesis equally makes use of underspecified locations.

3.2. Declarative Units

As pointed out in the last section, information holding at, or annotating, a tree node can be stated as declarative units[9], or tree node descriptions. Next to the treenode predicate, DUs most commonly include a formula (Fo) and a type (Ty) value:

9 The terminology Declarative Unit follows traditional LDSNL usage. Technically, tree node description (ND) is more appropriate.

(21) • {Tn(0), Fo(β(α)), Ty(t)}
/ \
• {Tn(00), Fo(α), Ty(e)} • {Tn(01), Fo(β), Ty(e -> t)}

The tree in (21) shows how information from the functor node combined with information from the argument node results in the complex formula value at the mother node. Similar to Categorial Grammar, application of modus ponens over type values is paralleled by function–application over formula values. Note, however, that LDSNL types are conditional types without an implication for the order of natural language expressions.

A tree like the one in (21) could be, for example, a simplified representation of an intransitive sentence:

(22) • {Tn(0), Fo(sing'(eve')), Ty(t)}
/ \
• {Tn(00), Fo(eve'), Ty(e)} • {Tn(01), Fo(sing'), Ty(e -> t)}

Formula values are representations of the meaning of words. The notational convention seen in (22) indicates an instruction to access a mental concept, for example, in the case of Fo(eve'), the mental concept the hearer has of Eve. Not all words encode instructions to access a named concept. Formula values of, for example, pronominal expressions include a meta-variable Fo(**u**), indicating an underspecified concept which encodes an instruction to the hearer to supply the formula value from the (cognitive) context, guided by Relevance considerations[10]. This search might be restricted, such as for the second person singular pronoun *you*, which encodes Fo($\mathbf{u}_{addressee}$), but the basic encoded meaning is here, as in the unrestricted case, an instruction for a search for a suitable conceptual representation. An example of a non-conceptual formula value is the encoded meaning of question pronouns, which encode a meta variable Fo(*WH*) possibly with a suitable restriction, e.g. +person, +thing, which is required to remain open in the eventual representation.

Type values consist of the elements {e, cn, t} and conditional types formed from these elements. The basic types stand for common noun, entity, and truth-value (i.e. a proposition) respectively. Conditional types are for example Ty(e -> t), as in the example above, or Ty(e -> (e -> t)) for transitive verbs.

10 One of the arguments against LF in LDSNL is that in VP ellipsis a suitable representation for pronominal expressions has to be established before the tree structure is completed (Kempson et al. 1999).

Other predicates found in DUs include Category (Cat) which indicates mood, for example in questions; Cat(+Q). This predicate is always associated at the mother node. Similarly, the representation of tense is expressed as values of predicates[11].

3.3. Requirements and Tasks States

The dynamics of the LDSNL system result from characterizing how trees grow. This means that LDSNL derivations, or parses, start from only minimal trees and proceed through stages of partial trees until the tree is fully developed. The minimal initial tree of a derivation is a tree with only a root node.

(23) • {Tn(0), ?Ty(t)}

The information holding at the root node at the outset of a derivation is, apart from the location, a *requirement* to derive an expression of Ty(t). This means that not only information already established can be stated in a DU, but also information which needs to be established. In this case, the requirement to derive an expression of Ty(t) reflects the justified expectation of a hearer that the tree building process will result in a (the) propositional form. Formally, the first step in a derivation can be expressed as an Axiom. At any stage in a derivation, some information might have been established, and some other information might still be required. Outstanding information is indicated by writing a question mark (?) in front of it. In the running text, a requirement can be identified by writing TODO, thus in the example above, there is a requirement TODO Ty(t) holding at Tn(0). The derivation is completed if, after all information from the lexicon has been incorporated into the tree, all requirements have been fulfilled.

For any stage in a derivation, a current node can be identified. The requirements holding at that node constitute the current task state:

```
(24)                         • {Tn(0), ?Ty(t)}
                                    / \
          • {Tn(00), Fo(eve'), Ty(e)}    • {Tn(01), ?Ty(e -> t)}
                                                / \
                    • {Tn(010), ?Ty(e) ◊}    • {Tn(011), Fo(see'),
                                               Ty(e -> (e -> t))}
```

11 An LDSNL treatment of tense and temporal dependencies is developed in Perrett (1996, fcmg.).

The parse state in (24) holds after the tree structure for *Eve saw* ... has been built. At this stage, there are three requirements, TODO Ty(t) at Tn(0), TODO Ty(e -> t) at Tn(01), and TODO Ty(e) at Tn(010). The pointer symbol ◊ indicates that the current node is Tn(010), so that the current task state is TODO Ty(e). If, in this situation, information from the lexicon provides an expression of Ty(e), it can be introduced into the tree at Tn(010), since it matches the current requirement. However, if the next word is *sing*, it cannot be introduced into the tree even though its type Ty(e -> t) matches the requirement holding at Tn(01). This is because the pointer is not at Tn(01).

3.4. Transition Rules

The development of tree structure involves the step from one parse state to another. This transition from one tree description to another can be licensed in two ways; either by a general transition rule, or by lexical instruction. I discuss transition rules first, and lexical instructions in the next section.

3.4.1. Introduction and Prediction

A transition rule specifies the input tree, that is at which state in a derivation it may apply, and an output tree, that is, what the resulting state is. Transition rules are stated as tree descriptions, where the input tree is described above the line and the output tree below the line:

(25) *Introduction*

$$\frac{\{\ldots ?Ty(Y) \ldots \Diamond\}}{\{\ldots ?Ty(Y),\ ?\langle d_0\rangle Ty(X),\ ?\langle d_1\rangle Ty(X \rightarrow Y), \ldots \Diamond\}}$$

Introduction licenses the introduction of modal requirements. If the current task is TODO Ty(X) (where X and Y are any well-formed type values), Introduction licenses the introduction of two modal statements to the effect that at the daughter nodes two subtasks are required which together result in Ty(Y). The rule licenses for example the introduction of the requirement for a subject when the variables are instantiated as follows:

(26) *Introduction (Subject)*

$$\frac{\{\ldots ?Ty(t)\ldots \Diamond\}}{\{\ \ldots ?Ty(t),\ ?\langle d_0\rangle\ Ty(e),\ ?\langle d_1\rangle\ Ty(e \text{->} t),\ \ldots \Diamond\}}$$

The input tree in (26) corresponds to the minimal tree at the outset of a derivation:

(27) • $\{Tn(0)\ ?Ty(t)\ \Diamond\}$

In this situation, Introduction results in:

(28) • $\{Tn(0)\ ?Ty(t),\ ?\langle d_0\rangle\ Ty(e),\ ?\langle d_1\rangle\ Ty(e \text{->} t)\ \Diamond\}$

The effect of this transition is that it introduces the statement that an expression of Ty(t) can be derived if there is an expression of Ty(e) at the argument daughter, and an expression of Ty(e -> t) at the functor daughter.

Modal requirements, such as those being introduced by Introduction in (28) result in the building of tree structure by the application of Prediction[12]:

(29) *Prediction*

$$\frac{\{_n \ldots ?\langle d_\pi\rangle\phi,\ \Diamond\}}{\{_n \ldots ?\langle d\rangle\ \phi \ldots\},\ \{_{n\pi} \ldots ?\phi,\ \Diamond\}}$$

where $\pi = 0$, or $\pi = 1$.

By Prediction, a new node is built. In a situation where at the current node a modal requirement holds, a new node can be built where this requirement minus the modal operator holds. This new node will then be the current node. By an application of Prediction, the tree in (28) can be developed into (30):

(30) • $\{Tn(0)\ ?Ty(t),\ ?\langle d_0\rangle\ Ty(e),\ ?\langle d_1\rangle\ Ty(e \text{->} t)\}$

/

• $\{Tn(00),\ ?Ty(e)\ \Diamond\}$

12 This is a slightly simplified version of the rule given in Kempson et al. (1999: 80), since I do not define Prediction for LINK relations (discussed below). The rule as given here is sufficient for the purposes of this thesis.

The new node Tn(00) is licensed since TODO $<d_0>$ Ty(e) holds at Tn(0). Since transition rules are in general optional, an alternative development of (28) would be (31):

(31) • {Tn(0) ?Ty(t), ?$<d_0>$ Ty(e), ?$<d_1>$ Ty(e –> t)}
\
• {Tn(01), ?Ty(e –> t) ◊}

Here Tn(01) is licensed since at Tn(0) the requirement TODO $<d_1>$ Ty(e –> t) holds. However, Prediction cannot apply to (28) twice, since after the first application, Tn(0) is no longer the current node.

3.4.2. Thinning, Elimination and Completion

While the rules discussed in the preceding section are concerned with unfolding of tree structure, the rules discussed in this chapter are concerned with the accumulation of established information.

The first rule provides a means for stating that requirements have been fulfilled:

(32) *Thinning*

$$\frac{\{\ldots\phi\ldots?\phi\ldots\Diamond\}}{\{\ldots\phi\ldots\Diamond\}}$$

The rule simplifies DUs. If at a current node a DU holds which includes both a fact and the requirement to fulfill this fact, the requirement can be omitted. The node is still the current node.

The rule of Elimination can be regarded as the opposite of Introduction:

(33) *Elimination*

$$\frac{\{\ldots <d_0> (Fo(\alpha), Ty(X)), <d_1> (Fo(\beta), Ty(X \text{–>} Y)) \ldots \Diamond\}}{\{\ldots Fo(\beta(\alpha)), Ty(Y), <d_0> (Fo(\alpha), Ty(X)), <d_1> (Fo(\beta), Ty(X \text{–>} Y)) \ldots \Diamond\}}$$

Elimination does not introduce a new node, but only changes annotations holding at one node. The rule states that if at a given node two modal statements hold. which state that both the argument daughter and the functor daughter are annotated with a formula and a type value, and the two type values can combine by Modus Ponens, then the resulting type and the corresponding expression derived by function–application over the formula values hold at that node. For example, for a tree describing *Eve sings,* Elimination would license the following transition:

(34) *Elimination* (Fo(sing'(eve')))

$$\frac{\{\ldots \langle d_0\rangle\ (Fo(eve'), Ty(e)), \langle d_1\rangle\ (Fo(sing'), Ty(e \rightarrow t)) \ldots \Diamond\}}{\{\ldots Fo(sing'(eve')), Ty(t), \langle d_0\rangle\ (Fo(eve'), Ty(e)), \langle d_1\rangle\ (Fo(sing'), Ty(e \rightarrow t)) \ldots \Diamond\}}$$

Elimination licenses the combination of information present as modal requirements. Elimination is similar to Introduction in that both rules specify transitions involving modal statements holding at a given node, without any direct effect on tree structure (as opposed to tree descriptions).

The rule of Completion can be regarded as the inverse of Prediction:

(35) *Completion*

$$\frac{\{_{\underline{n}} \ldots\}, \{_{\underline{n\pi}} \ldots \phi, \Diamond\}}{\{_{n} \ldots \langle d\rangle\phi, \Diamond\}, \{_{n\pi} \ldots \phi\}}$$

where $\pi = 0$ or $\pi = 1$.

Completion states that if at a daughter node some information holds, and the daughter is the current node, then that (i.e. mother) node may be annotated with the corresponding modal statement and becomes the current node. An example is given in (36):

(36)

$$\frac{\{_{\underline{0}} \ldots ?Ty(t), \langle d_{\underline{0}}\rangle\ (Fo(eve'), Ty(e)\}, \{_{\underline{01}} \ldots Fo(sing'), Ty(e \rightarrow t))\ \Diamond\}}{\{_{0} \ldots ?Ty(t), \langle d_0\rangle\ (Fo(eve'), Ty(e), \langle d_1\rangle\ (Fo(sing'), Ty(e \rightarrow t)\ \Diamond\}, \{_{01} \ldots Fo(sing'), Ty(e \rightarrow t))\}}$$

The input state in the example in (36) is a tree discription where at Tn(0) with the requirement TODO Ty(t) a modal statement holds to the effect that at the argument node Fo(eve') of Ty(e) holds (which in fact presupposes a previous application of Completion), and that at the (current) functor daughter Tn(01) Fo(sing') of Ty(e–> t) holds. The transition licensed by Completion is then that the presence of the information at Tn(01) is recorded at Tn(0) as a modal statement, and that the pointer moves to Tn(0), making it the current node. This new description is then in turn suitable input to Elimination.

The rules Elimination and Completion provide the means to pass established information up the tree so that eventually an expression of Ty(t) can be derived. Although the rules are optional, in practice they apply as late as possible, (ideally) when all terminal nodes are filled[13].

The rules introduced so far result, in conjunction with lexical instructions, in a complete parse. Modal requirements are introduced by Introduction, and the corresponding daughter nodes are built by Prediction. If lexical input fulfills a requirement, Thinning removes the requirement. After requirements have been fulfilled, Completion introduces modal statements at the mother nodes, and Elimination licenses the combination of formula values.

The next section introduces two more rules which are needed for more complex derivations.

3.4.3. Star Adjunction and Merge

The rules so far do not introduce unfixed nodes into the tree, or assign an unfixed node its place in the tree. The last two rules discussed in this section do this.

Star Adjunction licenses the introduction of an unfixed node at the outset of the parse:

(37) *Star Adjunction*

$$\frac{\{Tn(a),\ ?Ty(t),\ \Diamond\}}{\{Tn(a),\ ?Ty(t)\},\ \{\langle u^*\rangle Tn(a),\ ?Ty(e),\ \Diamond\}}$$

At a given parse where the current node Tn(a) has a requirement TODO Ty(t), the introduction of an unfixed node lower than Tn(a) is licensed, which becomes the new current node. The Star Adjunction rule is more specific than the rules discussed so far. It can only apply at the beginning of the parse, and it specifies that the new node be of Ty(e). This rule is needed to introduce fronted constituents such as question words into an unfixed node. Star Adjunction and unfixed nodes in general will be discussed more fully in Chapter 3.

---

13 More precisely, Elimination may apply after the tree is complete; the application of Completion is necessary at intermediate steps. This is shown in the sample derivation below, although in the remainder of the thesis, I generally assume that both Completion and Elimination provide the final steps in a derivation.

The last rule introduced here licenses the resolution of the underspecified node and assigns it to a fixed position. Merge licenses the merging of two node descriptions, i.e. DUs:

(38) *Merge*

$$\frac{\{\ldots ND, ND' \ldots\}}{\{\ldots ND \cup ND' \ldots\}}$$

The Merge rule is completely general. It simply states that two node descriptions can be combined into one. Although the rule overgenerates, it does ensure that an unfixed node can be merged with a fixed one, so that it is integrated into the tree.

The Merge rule concludes the discussion of transition rules in this section. The next section describes the structure of lexical entries.

3.5. Lexical Entries

The structure of lexical entries interacts with the general format of tree description introduced so far. Lexical information provides annotations on nodes and specifies how a particular lexical item contributes to the process of structure building. Lexical information interacts with the transition rules introduced in the last section in the process of tree building.

The general format for lexical information is as follows:

(39) *Format of Lexical Entries*

IF	Ty(X)
THEN	make(...),
	put (...),
	go (...),
ELSE	abort

Lexical entries consist minimally of three statements. The IF clause gives the condition under which the information provided by lexical entries can be introduced into the tree. For example, an expression of Ty(e) generally requires that there be a current requirement TODO Ty(e) at the stage at which the lexical entry is scanned. The THEN statement lists the particular actions which are performed if the condition in the IF statement is met. THEN statements consist usually of three predicates. The predicate make is an instruction to built a new node, which is further specified in the value of the predicate; for example make($<d_1>$) builds a new functor node. The predicate put is an instruction to annotate an existing node, for example with a formula and a type value. Finally, go is an instruction for pointer movement, it changes the current task state. Two details are important. First, make implies go, that is, the instruction to make a node implies that the pointer goes there. Second, the order of the predicates is important, for example put before make means put(...) at current node, then build a new node', while make before put means 'build a new node and put(...) there'. The ELSE clause specifies the actions to be performed if the IF condition is not met. Often this is abort, i.e. the end of the derivation, but the clause may be used otherwise. Finally, the clauses can be nested to result in more complex entries[14].

This characterization of lexical entries is a minimal characterization in the sense that all lexical entries which contribute to tree building contain at least the three clauses discussed. However, the actual actions performed may be more complex than outlined here, since more idiosyncratic information may be associated with individual words in the lexicon. This point is further discussed in relation to the analysis presented in Chapter 3.

A sample lexical entry for *love* is given below:

(40) *Lexical Entry for* love

IF	?Ty(e -> t)
THEN	put(?$<d_0>$ Ty(e)),
	make($<d_1>$), put(Fo(love'), Ty(e -> (e -> t)))
ELSE	abort

The condition for the introduction of the information from *love* is that the current task state is TODO Ty(e -> t). If this condition is met, the current node is annotated with a modal requirement. Then a new functor node is built and annotated with the formula and type values specified. If *love* is scanned at a stage in the derivation when the current task does not include the requirement TODO Ty(e -> t), the derivation ends. The entry for *love* is seen in use in the sample derivation in the next section.

14 The treatment of the lexicon here is compatible with arguments presented in Construction Grammar (cf. i.a. Fillmore 1988, Kay 1995) since lexical information includes (instructions to build) subtrees. In Construction Grammar, lexical entries may include larger syntactic structures (i.e. constructions) in which the lexical item is embedded.

3.6. Sample Derivation

In this section, I present a sample derivation of a simple sentence to show how the step-by-step process of tree growth is expressed in the LDSNL system. I display and comment on every step in the derivation, although I will give up this practice in the remainder of the thesis. The derivation here is merely meant to shows how the rules discussed above work.

The example to be discussed is given in (41):

(41) Sally loves chocolate.

The derivation begins with the introduction of the root node by Axiom:

(42a)

• {Tn(0), ?Ty(t) ◊}

At this stage, two rules may apply, Introduction and Star Adjunction. This is a situation which often arises, since transitions rules may apply optionally, and more than one rule might be applicable at any given stage of the parse[15]. In practice, I use the rules sensibly, so that in this case, Introduction applies:

(42b) • {Tn(0) ?Ty(t), $?\langle d_0 \rangle$ Ty(e), $?\langle d_1 \rangle$ Ty(e –> t) ◊}

By Prediction, the argument daughter can be built:

(42c)

• {Tn(0) ?Ty(t), $?\langle d_0 \rangle$ Ty(e), $?\langle d_1 \rangle$ Ty(e –> t)}

/

• {Tn(00), ?Ty(e) ◊}

At this stage the first word is scanned, namely *Sally*, for which I assume the following lexical entry:

(43) *Lexical Entry for* Sally

IF	?Ty(e)
THEN	put(Fo(sally'), Ty(e))
ELSE	abort

15 However, application of 'wrong' rules leads to a situation where the derivation cannot be completed. Recall that LDSNL does not model real time parsing, but characterizes the body of knowledge required for incrementally building interpretations.

The current task state matches the condition in the IF clause, so the formula value Fo(sally') and the type value Ty(e) can be introduced.

(42d)

• {Tn(0) ?Ty(t), ?<d_0> Ty(e), ?<d_1> Ty(e –> t)}
/
• {Tn(00), ?Ty(e), Fo(sally'), Ty(e) ◊}

At this stage, Thinning can apply to Tn(00) to remove the requirement:

(42e)

• {Tn(0) ?Ty(t), ?<d_0> Ty(e), ?<d_1> Ty(e –> t)}
/
• {Tn(00), Fo(sally'), Ty(e) ◊}

The DU at Tn(00) now matches the tree description which licenses the application of Completion:

(42f)

• {Tn(0) ?Ty(t), ?<d_0> Ty(e), ?<d_1> Ty(e –> t),
<d_0> (Fo(sally'), Ty(e)) ◊}
/
• {Tn(00), Fo(sally'), Ty(e)}

At this stage, two rules could apply at Tn(0) – Thinning, since one requirement holding at Tn(0) has been fulfilled, and Prediction, since there is still the modal requirement of the functor node. However, while Prediction could apply after Thinning, which does not move the pointer, Thinning could not apply after Prediction, which does move the pointer. Thus for clarity of display, Thinning applies first[16]:

(42g)

• {Tn(0) ?Ty(t), ?<d_1> Ty(e –> t), <d_0> (Fo(sally'), Ty(e)) ◊}
/
• {Tn(00), Fo(sally'), Ty(e)}

Now Prediction applies and builds the functor node:

(42h)

• {Tn(0) ?Ty(t), ?<d_1> Ty(e –> t), <d_0> (Fo(sally'), Ty(e))}
/ \
• {Tn(00), Fo(sally'), Ty(e)}   • {Tn(01), ?Ty(e –> t) ◊}

16 Alternatively, Thinning would apply after Completion.

The next step is the scanning of the word *love* which accesses the following information[17] (repeated here):

(40) *Lexical Entry for* love

IF ?Ty(e –> t)
THEN put(?$<d_0>$ Ty(e)),
make($<d_1>$), put(Fo(love'), Ty(e –> (e –> t)))
ELSE abort

The condition on the introduction of the lexical information from *love* is met since the current node has a requirement TODO Ty(e –> t). The first put statement annotates Tn(01) with a modal requirement, after which the make statement results in the building of a new functor node which is annotated with the information specified in the second put predicate:

(42i)
• {Tn(0) ?Ty(t), ?$<d_1>$ Ty(e –> t), $<d_0>$ (Fo(sally'), Ty(e))}
/ \
• {Tn(00), Fo(sally'), Ty(e)} • {Tn(01), ?Ty(e –> t), ?$<d_0>$ Ty(e)}
\
• {Tn(011), Fo(love'), Ty(e –> (e –> t)) ◊}

In this situation, Completion can apply to annotate Tn(01) with a modal statement registering the fulfilled requirement at Tn(011):

(42k)
• {Tn(0) ?Ty(t), ?$<d_1>$ Ty(e –> t), $<d_0>$ (Fo(sally'), Ty(e))}
/ \
• {Tn(00), Fo(sally'), Ty(e)} • {Tn(01), ?Ty(e –> t), ?$<d_0>$ Ty(e), $<d_1>$ (Fo(love'), Ty(e –> (e –> t))) ◊}
\
• {Tn(011), Fo(love'), Ty(e –> (e –> t))}

At this stage, Prediction applies at Tn(01) since there is a new modal statement, resulting from the lexical action from *love*. Thus Tn(010) can be built:

(42l)
• {Tn(0) ?Ty(t), ?$<d_1>$ Ty(e –> t), $<d_0>$ (Fo(sally'), Ty(e))}
/ \
• {Tn(00), Fo(sally'), Ty(e)} • {Tn(01), ?Ty(e –> t), ?$<d_0>$ Ty(e), $<d_1>$ (Fo(love'), Ty(e –> (e –> t)))}
/ \
{Tn(010), ?Ty(e) ◊} • {Tn(011), Fo(love'), Ty(e –> (e –> t))}

17 I omit tense and agreement throughout.

The next step is again scanning of lexical input, this time from *chocolate.* I assume here that mass nouns have at least one reading under which they are of Ty(e), so that the lexical information can be introduced into the tree:

(44) *Lexical Entry for* chocolate

```
IF      ?Ty(e)
THEN    put(Fo(chocolate'), Ty(e))
ELSE    abort
```

Since the current node requires a Ty(e) expression, the put statement can be applied:

(42m)

```
          • {Tn(0) ?Ty(t), ?<d1> Ty(e -> t), <d0> (Fo(sally'), Ty(e))}
                              / \
• {Tn(00), Fo(sally'),      • {Tn(01), ?Ty(e -> t), ?<d0> Ty(e),
  Ty(e)}                       <d1> (Fo(love'), Ty(e -> (e -> t)))}
                                     / \
          {Tn(010), ?Ty(e),     • {Tn(011), Fo(love'), Ty(e -> (e -> t))}
  Fo(chocolate'), Ty(e) ◊}
```

At this stage, all lexical information has been scanned and the verb's lexical requirements are fulfilled. All nodes have been built, so that the remaining steps serve only to combine the accumulated information. The only rules applying are thus Thinning, Completion and Elimination. The first step is the application of Thinning at Tn(010), followed by the application of Completion to Tn(010) and Tn(01). This results in an increase in modal statements at Tn(01):

(42n)

```
          • {Tn(0) ?Ty(t), ?<d1> Ty(e -> t), <d0> (Fo(sally'), Ty(e))}
                              / \
• {Tn(00), Fo(sally'),      • {Tn(01), ?Ty(e -> t), ?<d0> Ty(e),
  Ty(e)}                       <d1> (Fo(love'), Ty(e -> (e -> t))),
                               <d0> (Fo(chocolate'), Ty(e)) ◊}
                                     / \
          {Tn(010), Ty(e),      • {Tn(011), Fo(love'), Ty(e -> (e -> t))}
          Fo(chocolate')}
```

Next, Thinning applies at Tn(01). I only reproduce Tn(01), since the rest of the tree remains unchanged:

(42o)

```
• {Tn(01), ?Ty(e -> t), ?<d0> Ty(e),
<d1> (Fo(love'), Ty(e -> (e -> t))),
<d0> (Fo(chocolate'), Ty(e)) ◊}
```

The Ty(e) expression at the argument node fulfills the requirement TODO $\langle d_0 \rangle$ Ty(e), which is removed. Elimination then applies to the values of the two daughter nodes:

```
(42p)    • {Tn(01), ?Ty(e -> t), (Fo(love'(chocolate')), Ty(e -> t)),
         <d1> (Fo(love'), Ty(e -> (e -> t))),
         <d0> (Fo(chocolate'), Ty(e)) ◊}
```

The derived fact Ty(e –> t) fulfills the requirement TODO Ty(e –> t), which is removed:

```
(42q)      • {Tn(0) ?Ty(t), ?<d1> Ty(e -> t), <d0> (Fo(sally'), Ty(e))}
                              / \
         • {Tn(00), Fo(sally'),     • {Tn(01),
           Ty(e)}                      (Fo(love'(chocolate')), Ty(e -> t)),
                                       <d1> (Fo(love'), Ty(e -> (e -> t))),
                                       <d0> (Fo(chocolate'), Ty(e)) ◊}
                                     / \
               {Tn(010), Ty(e),      • {Tn(011), Fo(love'), Ty(e -> (e -> t))}
               Fo(chocolate')}
```

Completion applied to Tn(0) and Tn(01) annotates Tn(0) with a modal statement. I represent only Tn(0):

```
(42r)    • {Tn(0) ?Ty(t), ?<d1> Ty(e -> t), <d0> (Fo(sally'), Ty(e)),
           <d1> ((Fo(love'(chocolate')), Ty(e -> t)) ◊}
```

By Thinning and Elimination the derivation ends with the final tree in (42s):

```
(42s)      • {Tn(0), Ty(t), Fo(love'(chocolate')(sally')),
           <d1> (Fo(love'(chocolate')), Ty(e -> t)), <d0> (Fo(sally'), Ty(e)) ◊}
                              / \
         • {Tn(00), Fo(sally'),     • {Tn(01),
           Ty(e)}                      (Fo(love'(chocolate')), Ty(e -> t)),
                                       <d1> (Fo(love'), Ty(e -> (e -> t))),
                                       <d0> (Fo(chocolate'), Ty(e)) ◊}
                                     / \
               {Tn(010), Ty(e),      • {Tn(011), Fo(love'), Ty(e -> (e -> t))}
               Fo(chocolate')}
```

In the final tree, the last oustanding requirement, TODO Ty(t) at the root node has been fulfilled by the derivation of the corresponding formula value Fo(love'(chocolate')(sally')). The derivation shows that structure building is achieved by the rules Introduction and Prediction in conjunction with lexical information, while the combination of information in the tree is achieved by the rules Thinning, Completion and Elimination.

In the remainder of this thesis, I will display derivations in much less detail, and concentrate on the growth of tree structure, as opposed to the combination of information. I will also in general omit modal statements and pointer movement, unless necessary.

3.7. Displacement Structures

As already indicated, displaced constituents such as question words and dislocated topics, are analysed by employing the Kleene star operation over tree modalities and the corresponding unfixed nodes. In this section I present the relevant steps in a derivation of (45):

(45) What does Sally love?

I ignore the contribution of the auxiliary and just concentrate on basic predicate–argument structure. I also ignore irrelevant details.

The initial step is as always derived by Axiom:

(46a) • {Tn(0), ?Ty(t)}

In contrast to the derivation above, the first step is here not Introduction, but Star Adjunction:

(46b) • {Tn(0), ?Ty(t)}

{Tn(0*), ?Ty(e)}

The relevant (abbreviated) lexical entry for *what* is given below:

(47) *Lexical Entry for* what

IF	{$_{n^*}$?Ty(e)}
THEN	put(Fo(WH), Ty(e)
	go(<u*>), put(+Q)
ELSE	abort

The IF clause requires a TODO Ty(e) task to hold at an unfixed node. Since this condition is met, the node is annotated with the formula and type values given. The pointer then moves to the root node and puts a +Q feature to indicate that the proposition is a question:

```
(46c)                    • {Tn(0), Cat(+Q) ?Ty(t) ◊}

          {Tn(0*), Fo(WH), Ty(e)}
```

At this stage, Introduction can apply at Tn(0) and the derivation proceeds as in the sample derivation above, plus the presence of the unfixed node, until the information from *love* has been incorporated into the tree:

```
(46d)                      • {Tn(0), Cat(+Q), ?Ty(t)}
                                    / \
          • {Tn(00), Fo(sally'),      • {Tn(01), ?Ty(e -> t)}
             Ty(e)}                     /\
                           {Tn(010),   • {Tn(011), Fo(love'), Ty(e -> (e -> t))}
                            ?Ty(e) ◊}

          {Tn(0*), Fo(WH), Ty(e)}
```

At this stage, there is no further lexical input, but there is still a requirement TODO Ty(e) outstanding, namely at Tn(010), and there is an unfixed node. A derivation is not completed unless all nodes have a fixed position in the tree. In this case, the underspecified node can be identified with the fixed node at Tn(010) by Merge, where the requirement TODO Ty(e) can be fulfilled by the type value of the pronoun. The underspecified location can thus be resolved as Tn(0*) = Tn(010), and the eventual tree results:

```
(46e)                    • {Tn(0), Cat(+Q), ?Ty(t)}
                                    / \
          • {Tn(00), Fo(sally'),      • {Tn(01), ?Ty(e -> t)}
             Ty(e)}                     /\
                           {Tn(010),   • {Tn(011), Fo(love'), Ty(e -> (e -> t))}
                      Fo(WH), Ty(e)}
```

In this tree, all nodes are fixed, so that Completion and Elimination can apply.

The analysis of displaced constituents thus exploits the possibility to have underspecified tree node locations, which results in an unfixed node during the derivation, which only eventually is assigned a fixed place in the tree, before Completion and Elimination apply.

3.8. LINKed Structures

The LINK relation constitutes an extension of the LDSNL system as discussed so far. A LINK relation can be established between two trees whose nodes then do not stand in a daughter or unfixed daughter relation. A LINK analysis is employed in LDSNL for relative clause:

(48) Sally, who I admire, is leaving tomorrow.

The relative clause is in a LINK relation to the head noun, which means that it is an island for extraction (it is not a daughter relation – an unfixed node in the matrix clause cannot be fixed into a LINKed structure), but yet part of the tree in the sense that a copy of the formula value of the head noun has to be incorporated into the LINKed tree. The LINK relation is established as follows:

(49) *LINK Induction*

$$\frac{\{_{n} \ldots Fo(\alpha), Ty(e) \ldots \Diamond\}}{\{_{n} \ldots Fo(\alpha), Ty(e) \ldots\}, \{_{nL0} \ldots ?Ty(t) \ldots \Diamond\}, \{_{nL0^{*}} \ldots Fo(\alpha), Ty(e) \ldots\}}$$

The rule states that from a Ty(e) expression with some formula value, a transition is licensed in which two new LINKed nodes are built; a new root node with TODO Ty(t), and new unfixed node below the new LINKed root node with the type and formula value being identical to those of the head noun. The corresponding tree structure is as follows:

(50)
```
                      • {Tn(0), ?Ty(t)}
                              / \
• {Tn(00), Fo(sally'),          • {Tn(01), ?Ty(e -> t)}
          Ty(e)}
             \
            LINK
               \
          • {Tn(00L0), ?Ty(t) ◊}

{00L0* Fo(sally'), Ty(e)}
```

From this tree, the derivation proceeds standardly with the building of the relative clause. The unfixed node behaves like the question pronoun discussed in the last section. After the due steps, subcategorization information from the verb *admire* leads to the building of a node with a requirement TODO Ty(e), at which the unfixed node can be fixed. The derivation then continues with the development of the matrix clause[18].

The LINK relation has been developed further by Swinburne (1999) in an analysis of secondary predication. In this thesis, I do not discuss relative clauses, but secondary predication is briefly discussed in Chapter 3.

4. Conclusion and Outline of the Thesis

In this chapter, I have outlined the theoretical background for the analysis being developed in this thesis. In the first half of the chapter, I have discussed the conceptual assumptions of LDSNL, and introduced a model of utterance interpretation in which the LDSNL model is embedded. I have discussed work in Government Phonology and Relevance Theory which supports the view that knowledge of language can at least partly be characterized as the underlying ability of hearers to assign representations of meaning to an incoming physical speech signal. Within this overall outline, I have characterized the LDSNL approach to natural language syntax and syntactic knowledge. From the perspective of LDSNL, words provide the hearer with instructions to incrementally develop a structured representation of the propositional form of the utterance. This representation is built from lexical and syntactic information on the one hand, and non-demonstrative pragmatic inference on the other. Model-theoretic semantic interpretation is only assigned to the propositional form.

In the second half of the chapter, I have introduced the formal tools employed in LDSNL to model a process of tree building by which the conceptual claims are formally articulated. I have introduced basic LOFT tree structures and declarative units which annotate trees, as well as the formal definitions which characterize the dynamic aspect of tree building, that is, task states and requirements, transition rules and lexical actions. Finally, I have briefly introduced the LDSNL analyses of displaced constituents involving unfixed tree nodes, and of relative clauses involving the LINK relation. In the remainder of this section, I give an outline of the thesis.

The following Chapter 2 introduces the empirical problem discussed in the thesis, namely the problem of verb phrase adjunction. A range of cross-linguistic data as found in the literature is introduced which shows that the distinction between arguments and adjuncts is not clear-cut, indicating that at

18 The relevant pointer movement has to be added to the LINK rule above. Cf. the full treatment in Kempson et al. (1999: Chapter 4), and Swinburne (1999).

some level of analysis, obligatory and optional nominal constituents of the verb phrase have to be assigned identical structural relations to the verb. The main syntactic evidence for this claim comes from extraction patterns, where arguments and adjuncts can be shown to behave alike. The view is further supported by semantic and morphological considerations. The chapter delimits the data to be discussed in the thesis, which include NPs and PPs, but only marginally lexical adverbs, and furthermore only those PPs and NPs which function as, possibly optional, arguments of the verb, to the exclusion of sentence and NP adverbials. In light of the evidence presented, problems for an LDSNL analysis of adjuncts are discussed and possible solutions are offered.

With both theoretical background and empirical range determined, I come to the main proposal made in the thesis in Chapter 3. I propose that verb phrase adjunction in LDSNL can best be analysed by introducing an underspecified semantic type for verbs which licenses the optional introduction of Ty(e) expressions (i.e. NPs and PPs) as arguments of the verb. The type specification employs the Kleene star operation, so that verbal underspecification can be explicitly encoded. This analysis of adjunction – which I call the e* analysis following the definition of the underspecified type – is developed as an alternative to a putative adjunction analysis which locates the underspecification inherent in verb phrase interpretation in the adjunct rather than in the verb. I show that the e* analysis has a number of advantages over underspecified adjuncts. The main part of the chapter then shows how the underspecified type is integrated into the LDSNL system, and how verb phrase structures are built with it. All necessary formal specifications are given in this chapter, including transition rules and relevant lexical entries. The e* analysis entails an analysis of PPs as being of Ty(e) expressions, and the necessary formal analyses of PPs and of prepositions are developed. The chapter is concluded by discussing a number of implications and consequences of the e* analysis.

Chapter 4 turns to the question of the semantic interpretation of underspecified verbs. Three main approaches to the semantics of adjuncts proposed in the literature are discussed – an analysis under which adjuncts are functors taking VPs as arguments (Dowty 1979), an analysis which treats adjuncts as arguments to the verb (McConnell-Ginet 1982), and an analysis which introduces underspecified semantic relations (Minimal Recursion Semantics, Copestake et al. 1997). After some discussion, I show that the approach developed by McConnell-Ginet is the most suitable one for the interpretation of e* verbs, and I develop an incremental extensional semantics for underspecified verbs based on this work. The chapter thus also serves to illustrate how the proposal developed in this thesis relates to previous work on adjunction. A concluding discussion is offered, which provides the transition to the next chapter.

The semantic characterization of e* verbs is based on the tacit assumption that model-theoretic semantic interpretation is assigned to the natural language string directly, an assumption inherited from the analysis proposed by McConnell-Ginet. However, LDSNL assumes, following Relevance Theory, that propositional content to which a model-theoretic interpretation can be assigned is only arrived at by processes of pragmatic inference. Chapter 5 explores the role of pragmatic reasoning in the interpretation of underspecified verbs. It is argued that in particular the Relevance theoretic analysis of mental concepts and processes of concept formation (e.g. Carston 1996, Sperber & Wilson 1997) can be employed for e*. The basic assumption of this analysis is that words encode only incomplete concepts, which are in need of pragmatic enrichment in order to play a role in the derivation of inferential effects. From this perspective, the underspecification of verbs as developed in the e* analysis can be viewed as being simply an overt syntactic reflex of this much more general process. Verbs encode their incomplete conceptual meaning by being syntactically underspecified with respect to the number of Ty(e) expressions with which they combine on a given occasion. Complementarily, optional Ty(e) expressions can be regarded as an aid in concept formation. This analysis confirms the view that natural language understanding is subject to pragmatic inferencing throughout. Some further implications of this analysis are discussed.

Chapter 6 presents an analysis of applied verbs in Swahili. Applied verbs have often been characterized as being related to a corresponding base verb by an operation of valency changing, so that, for example, a transitive verb becomes di-transitive. In this chapter I develop an alternative hypothesis which assumes that applied verbs encode an instruction for concept formation, so that the hearer is entitled to derive additional contextual effects. I present evidence to support this view, and show how change in valency follows from the unified analysis presented.

In Chapter 7 I explore the possibility of how the analysis of underspecified verbs developed here could be used in a computational implementation of LDSNL. I introduce two approaches which model context-sensitive reasoning with natural language input, Generative Lexicon Theory (Pustejovsky 1995), which employs typed feature structures, and the logic based approach described in Hunter & Marten (1999), and I show that the second approach is more suitable for modelling reasoning with underspecified verbs. At this stage, no fully worked out analysis is available, so that the discussion in this chapter is only a preliminary exploration of possibilities.

The final Chapter 8 provides the conclusion of the thesis, summarizes its results and offers a brief evaluation.

Chapter 2

Arguments and Adjuncts

1. Introduction

In this chapter I introduce the main topic of the thesis, namely the relation between verbs and their complements in the verb phrase. I start by discussing the notions of verb phrase and subcategorization and give an LDSNL characterization of these terms (Section 2). Section 3 presents a range of examples which show that the distinction between arguments and adjuncts, which is implied by the notion of strict subcategorization, mainly reflects the difference between obligatory and optional constituents of the verb phrase, but that the distinction is not clearly supported by morphological, semantic, or syntactic facts. In Section 4, I discuss how, in the light of the evidence presented, an LDSNL analysis of adjunction can be developed. The section serves as a preparatory discussion of the analysis developed in the following chapters and introduces the particular problems for tree growth and subcategorization raised by adjuncts. Section 5 summarizes the findings of this chapter.

2. Verb Phrase and Subcategorization

In this section I provide an LDSNL characterization of the notion of verb phrase and subcategorization. Since in particular subcategorization plays an important role in the following chapters, this section includes a brief discussion of several approaches to model subcategorization requirements, against which the LDSNL analysis will be contrasted. An important point for the following sections is that strict subcategorization implies a distinction between obligatory and optional constituents of the verb phrase, i.e. between arguments and adjuncts.

2.1. Verb Phrase

There is no primitive notion of verb phrase in the LDSNL system, i.e. there is no rule, as for example found in phrase structure grammars, which specifies how a VP is built, nor is there a primitive notion of verb. Similar to Categorial

Grammar, verbs are analysed as predicates, which specify the number of nominal expressions with which they combine to yield a proposition. For example, a transitive verb may be associated lexically with a type specification such as (1):

(1) *Type Specification for a Transitive Verb*

Ty(e -> (e -> t))

The type in (1) indicates that two Ty(e) expressions are required by the predicate to result in an expression of Ty(t). However, there is one important difference between LDSNL and Categorial Grammar with respect to typing, namely that LDSNL has no general rules of type inference such as type shifting or function–composition[19]. This means that an LDSNL derivation with a transitive verb always proceeds by first combining the verb with the object, and then combining the verb plus object with the subject. If type shifting were allowed as one of a number of type inference rules, the subject could be assigned a higher type such as (2):

(2) *Potential Type for Type-Shifted Subject*

Ty((e -> t) -> t)

The type in (2) would by function composition license a derivation where the subject acts as a functor which takes a transitive verb as an argument, and where the subject plus verb then combines with the object to yield an expression of Ty(t). But this means that there is no means to structurally define a VP, i.e. an expression which includes the verb and its object(s), but not the subject. It is because there are no type inference rules in LDSNL that the notion of verb phrase can be defined structurally, namely as an expression of Ty(e -> t), where, as in (2), the Ty(e) expression is the subject. The corresponding notion of subject can thus be defined as that Ty(e) expression which combines with a predicate to result (immediately) in an expression of Ty(t). The corresponding tree relations are as follows:

19 See e.g. Lambek (1958), McGee Wood (1993), Morrill (1994).

(3)

```
                • {Tn(0), Ty(t)}
                      /\
• {Tn(00), Ty(e)}   • {Tn(01), Ty(e -> t)}
                              /\
      • {Tn(010), Ty(e)}   • {Tn(011), Ty(e -> (e -> t))}
```

In the tree in (3), the expression at Tn(00) corresponds to the subject. A verb phrase is an expression holding at Tn(01). Constituents of the verb phrase can thus be characterized as Ty(e) expressions holding below Tn(01). It is in the sense of this characterization that I employ the term verb phrase in this thesis.

2.2. Subcategorization

Type information such as in (1), above, is used to specify the valency, or transitivity of verbs. That is, LDSNL assumes, following traditional grammar, that verbs specify in the lexicon how many nominal expressions can be associated with it in a clause, from which classifications of verbs into, for example, intransitive, transitive, and di-transitive verbs can be derived. Typing information is also used to distinguish between arguments and adjuncts: the former are the subcategorized NPs, while the latter are not, or at least not directly, licensed by the verb, so that their presence in a well-formed clause is regulated by some other principle of the grammar. For example, (4) shows that a putative lexical entry for *kiss* includes a statement to the effect that *kiss* both licenses and requires a subject NP and an object NP:

(4) kiss, Ty(e -> (e -> t))

The subcategorization information associated with the verb accounts for the fact that only (5a) is grammatical, while in (5b) the object is missing, and in (5c), there is one NP too many, so that both these sentences are ungrammatical:

(5a) Jill kissed Robert.
(5b) *Jill kissed.
(5c) *Jill kissed Robert John.

It also follows from (4) that the PP *in the garden shed* in (6) is an adjunct, and thus not directly licensed by the verb:

(6) Jill kissed Robert in the garden shed.

In the following section, I briefly give a more general overview of how subcategorization information can be expressed, and then show in more detail how subcategorization information is expressed in LDSNL.

2.2.1. Means of Expressing Subcategorization

In addition to the syntactic aspects of subcategorization information, there are semantic considerations. Thus, for example, in Categorial Grammar syntax and model theoretic semantics are defined in tandem in analogy to the syntax and semantics of logical languages. From a Categorial Grammar perspective, the fact that *kiss* syntactically categorizes for two NPs follows from the fact that kiss is a binary predicate in some model in which it receives its interpretation. That is, in addition to syntactic subcategorization, there is, in Categorial Grammar, semantic subcategorization. Thus, for example, a statement about syntactic subcategorization such as in (4), above, corresponds to the statement (7) in the semantics:

(7) λxλykiss(y,x)

As stated in (7), the predicate kiss combines with two expressions which are substituted for the variables to result in a proposition such as, for example, shown in (8):

(8) kiss(jill, robert)

Accordingly, ungrammatical sentences like (5b) and (5c) above are incomplete or ill-formed in the semantics, since they do not result in an expression of Ty(t).

Frameworks not, or less, based on logic often employ the term participants for the semantic corrrelate to subcategorization. The idea here is that verbs have the valency they have because they refer to some semantic entity, like for example a frame, event, or scenario which involves the number of participants which are identified as subcategorized NPs by the verb. Despite considerable differences between more formal and more functional approaches to natural language analysis, in this particular respect, both are fairly similar – the syntactic restriction on NPs encoded by the verb is, at least partly, explained with reference to the meaning the verb expresses, however this meaning is represented.

There is a third alternative to characterize the relation between syntactic transitivity and semantic arity or participants, namely to employ thematic roles (or, theta(θ)-roles, deep cases). Thematic roles can be viewed as expressions in an intermediate vocabulary expressing generalizations over both conceptual (and/or semantic) structure and syntax. The major motivation for thematic roles is that they make it possible to characterize discrepancies between syntax and semantics, as well as cross-linguistic variation as to which semantic argument is expressed by which syntactic argument. For example, in GB theory (Chomsky 1981), both theta-theory and case-theory regulate the occurrence of nominal expressions; correspondingly, verbal lexical entries include information about both thematic roles and case. LDSNL, following the Categorial Grammar tradition, does not recognize this intermediate level and does not employ thematic roles. I therefore offer only a short discussion here, without attempting to review the considerable literature on thematic structure even in outline[20].

The main reason for rejecting thematic roles as part of the analytical apparatus is the vagueness of that concept. Specifically, there is, first, no set of well-defined theta-roles, and, secondly, their status as syntactic or semantic primitives is unclear. The first point, that there is no well-defined set of thematic roles, has often been made. While most frameworks or analyses would include roles like agent and patient, other potential members like for example source, path, goal, location, or direction are less firmly established. Furthermore, it is not clear what the exact distinction between the (or some) members of a given set of thematic roles is, nor is it always clear which role to assign to a given noun phrase in a given natural language expression. The more technical reason for excluding thematic roles on these grounds is that all expressions in the LDSNL model have an operational semantics, which might, however, be difficult to formulate for thematic roles. At present, thematic roles are thus not part of the LDSNL system.

The second point, the unclear status of thematic roles with respect to syntax or semantics is apparent in the following two quotes: "... in GB the focus is on the semantic relations holding between heads and their syntactic complements. These relations, called *thematic roles*, are stored in the lexical entries of potential heads ..." (Ravin 1990: 3, emphasis in the original), and "... the notion 'θ-grid' is no more a semantic notion than θ-theory is a semantic theory" (Tomaselli 1997: 144). That is, on the one hand, thematic roles can be

20 Cf. Campe (1994) for references. The following discussion is based a.o. on Ravin (1990), Ladusaw & Dowty (1988), and the papers in Butt & Geuder (1998).

viewed as syntactic primitives, purely technical tools to determine syntactic well-formedness, while on the other hand, names like agent or experiencer seem to imply that these roles refer to something more semantic or conceptual. Assuming that this indeterminate status is undesirable, one can divide the question of thematic roles into two parts; syntactically, they are not part of the LDSNL system due to the difficulty in defining them, while semantically, they may be regarded as generalizations over mental representations, or concepts, but not as primitives (cf. Ladusaw & Dowty 1988). I thus assume that thematic roles play no role in the specification of verbal subcategorization in LDSNL.

2.2.2. Subcategorization in LDSNL

The analysis of subcategorization in LDSNL is similar to the Categorial Grammar approach, in that there is no notion of participants, or thematic roles, and in that typing information is taken to reflect semantic arity. LDSNL types can thus be defined in correspondence with lambda expressions as in Categorial Grammar, as for example in (9):

(9) kiss: {Fo(λxλy**kiss**(x)(y)), Ty(e –> (e –> t))}

In (9) the conditional type corresponds to two steps of lambda reduction which are needed to derive a proposition.

There is however an important conceptual difference between Categorial Grammar and LDSNL with respect to the relation between syntactic structure and semantic interpretation. In LDSNL, there is considerable independence of the building of semantic trees and eventual model-theoretic evaluation, whereas in categorial grammar every operation in the syntax corresponds to an operation in the semantics. Since in LDSNL the process of tree growth is defined syntactically, independent of the eventual semantic evaluation, there is no (model-theoretic) semantic correlate for tree-underspecification, or any model-theoretic semantic interpretation for procedural aspects of lexical items such as *wh*-pronouns. For the same reason, subcategorization statements in LDSNL are not so much concerned with how eventual semantic evaluation in a model is reflected, but rather with the contribution of lexical items to the process of structure building. In addition to type information, lexical entries for verbs may thus include explicit instructions about the transitions licensed by the verb. As shown in Chapter 1, a transitive verb like *kiss* may lexically specify the following actions:

(10) *Lexical Entry for* kiss

IF	?Ty (e -> t)
THEN	put(?<d_0> Ty(e)),
	make(<d_1>), put(Fo(kiss'), Ty(e -> (e -> t))
ELSE	abort

The lexical entry does not only specify the type and formula value of the verb, but rather consists of a set of actions which are licensed by the lexical entry. In this case, the information from the verb results not only in the building of the verb's own node (the functor daughter), but also in the annotation of the VP node with a modal requirement, which effectively, given the availabilty of Prediction, licences the building of the argument node as well. Lexical entries in LDSNL do not only include the subcategorization requirements of a given verb, but in addition include instructions for the establishment of corresponding tree structure. Subcategorization plays thus an important part in the building of tree structure.

In view of this characterization, the relation between type information and eventual model theoretic interpretation can be regarded as being mediated by the process of structure building, as well as possibly other processes relevant for the establishment of interpretation. The exact relation between the type information provided by verbs and their eventual semantic interpretation constitutes one of the main topics of the thesis, fully developed in the following chapters. In particular, I argue that verbal subcategorization determines only partly the arity of a given predicate, but that quite generally, information from words is subject to pragmatic enrichment, so that the eventual semantic representation of a verb is only determined in context. In order to express the possibility of structural and contextual contribution to the establishment of meaning in utterance interpretation, I follow the LDSNL convention to represent formula values not as lambda expressions, but by writing, for example, Fo(kiss'), which indicates that the word *kiss* provides an instruction to the hearer to access the concept **kiss**. It is this concept, rather than the instruction to access it, which can be characterized as being of a particular arity. Thus, in practice, the number of Ty(e) expressions with which a given verb can combine is stated in the type value, but not explicitly in the formula value, at least not unless necessary.

2.3. Summary

The basic LDSNL notions introduced in this section are thus verb phrase, which can be characterized as tree structure holding below a node with a Ty(e -> t) predicate, and subcategorization, which can be seen as involving type and formula values, but more importantly includes instructions for building of tree structure, since lexical entries may include sets of actions which are performed when the verb is introduced into the tree. In the following sections I discuss the relation between subcategorization and verb phrases, that is, the distinction between obligatory and optional constituents of the VP. The discussion serves to introduce the empirical range of the analysis of verbal subcategorization proposed in this thesis.

3. Arguments and Adjuncts

One of the implications of a traditional view of subcategorization is that there is a clear distinction between arguments and adjuncts. The former are subcategorized, necessarily expressed nominal expressions, while the latter are optional, not always nominal expressions in a more loose relationship to the verb. This distinction correlates to some extent with a difference in semantic function and morphological marking. Semantically, arguments introduce the main participants of the event or action denoted by the verb, while adjuncts add further information which is not strictly speaking necessary – about place, time, purpose, instruments used, or other people involed. Morphologically, arguments are marked with nominative (or ergative) and accusative case, or simply unmarked, while adjuncts are marked with a non-core case, introduced by a preposition, or marked as adverbs. However, as has been discussed by inter alia McConnell-Ginet (1982) and Fillmore (1994), this correlation is not perfect, and the distinction between arguments and adjuncts might therefore not be as clear as implied by strict subcategorization.

3.1. Morphological Marking

Leaving the semantics to one side for the moment, the argument–adjunct distinction is imperfect in the sense that a number of verbs require morphological adverbs as obligatory, rather than optional complements. Consider the following examples:

(11a) John behaved rudely to his mother in law.
(11b) The chancellor worded the bill carefully.
(11c) The secretary phrased the submission badly.

The adverbs in (11) are clearly morphologically marked as such, but seem to be obligatorily required:

(12a) ?John behaved to his mother in law.
(12b) ?The chancellor worded the bill.
(12c) ?The secretary phrased the submission.

The sentences in (11) without the adverb are less well-formed than the corresponding sentences with adverbs, so that the adverbs appear to be subcategorized for by the verb.

Similarly, in (13) the prepositional phrases cannot be omitted:

(13a) Fran put the kettle on the stove.
(13b) Alex lay the book on the table.
(13c) Judy lives with her sister.
(13d) Donavan resides in Oyster Bay.

(14a) *Fran put the kettle
(14b) *Alex lay the book
(14c) ?Judy lives
(14d) ?*Donavan resides

As these examples show, the PPs in (13) behave as arguments rather than as adjuncts with respect to obligatoriness.

Finally, the examples in (15) show obligatory verb phrase constituents in German which are not accusative marked:

(15a) *Axel half dem Jungen*
A helped the.DAT boy.DAT
'Axel helped the boy'

(15b) *Sie gedachten der alten Freunde*
they remembered the.GEN old.GEN friends.GEN
'They remembered the old friends'

The object in (15a) is dative case marked, while the one in (15b) is genitive marked. Although the canonical object case in German is accusative, the NPs in (15) cannot be omitted:

(16a) *?Axel half*
(16b) **Sie gedachten*

As in the preceding examples, the NPs here appear to be arguments, despite the fact that they are not marked with canonical argument case.

All the preceding examples show that the distinction between arguments and adjuncts is not co-extensive with any morphological distinction, so that neither the notion of argument, nor the notion of adjunct can simply be defined in terms of morphological coding.

3.2. Semantic Function

From the point of view of subcategorization, adjuncts provide additional, or circumstantial information, as opposed to core information provided by arguments of the predicate. For adverbial modification in general, a distinction is often made between sentence adverbs, verb phrase adverbs and verb adverbs:

(17a) He will probably be late.
(17b) Jill put down the ring reluctantly.
(17c) He covered himself completely.

The sentence adverb in (17a) adds information about the likelihood of the proposition expressed by the sentence. In (17b), the adverb appears to modify the verb phrase, while the adverb in (17c) seems to modify the verb. In addition, PPs functioning as adverbials can, in contrast to adverbs, modify a nominal expression:

(18a) I always liked to talk to the boys from South Dakota.
(18b) The man with the paper has just left.

In (18a) the PP *from South Dakota* modifies the NP *the boys,* and in (18b) the PP *with the paper* modifies the NP *the man.* The difference between PPs which modify NPs and those which function as complements to verbs will be discussed in more detail in Chapter 3.

On the other hand, PPs often resist a sentence adverbial reading available to an adverb, as shown in (19) and the parallel German example in (20):

(19a) Carefully, David had cut all the bagels.

(19b) With care, David had cut all the bagels.

(20a) *Julia stellte die Blumen sorgfältigerweise auf den Tisch*
Julia put the flowers carefully on the table
'Julia put the flowers carefully on the table'

(20b) *Julia stellte die Blumen mit großer Sorgfalt auf den Tisch*
Julia put the flowers with great care on the table
'Julia put the flowers with great care on the table'

While (19a) may have a reading 'it was careful of David ...', (19b) can only mean that the act of cutting was done with care. Similarly, (20a) may mean that it was careful of Julia to put the flowers on the table, while (20b) can only mean that she did it carefully.

In this thesis I am mainly concerned with PPs which appear to be part of the verb phrase, as opposed to those functioning as sentence adverbial.

When modifying verbs or verb phrases, PPs express a variety of semantic 'functions', some of which are parallel to the semantic function of arguments as illustrated below:

Locative

(21a) Jane was singing in the bathroom.

(21b) Judy lives in Notting Hill.

Directional

(22a) Olli is leaving for Tokyo tomorrow.

(22b) Sara put the pizza into the oven.

Temporal

(23a) I have to leave before midnight.

(23b) The meeting lasted six hours.

Commitative

(24a) Sally went to the movies with James.

(24b) I am staying with my parents.

The list could be expanded or refined, but it shows that the common characteristic of all these examples is that the PPs in the (a) sentences modify the action denoted by the verb, rather than the proposition expressed by the sentence, in a manner similar to the subcategorized objects in the (b) sentences, so that there does not seem to be any obvious well-defined distinction between arguments and adjuncts on semantic lines.

Another semantic parallel between arguments and adjuncts has been noted by Tenny (1994), namely that both arguments and adjuncts may change the aspectual information of the verb:

(25a) Sally painted for/*in an hour.
(25b) Sally painted the picture *?for/in an hour

(26a) Rob pushed the table for/*in an hour.
(26b) Rob pushed the table to the window *?for/in an hour.

Both the object *the picture* in (25) and the PP *to the window* in (26) change the aspectual status of the predicate from non-telic to telic, as is indicated by the temporal adverbial. I do not discuss aspect in detail here, but just note that arguments and adjuncts can have identical semantic functions with respect to aspectual information.

It is examples like those in (21 – 26) with which I am mainly concerned in the thesis, i.e. PPs which modify the action denoted by the verb. The relation between the analysis developed here and VP modification by adverbs and nominal adverbials (e.g. *yesterday*) is only discussed briefly at relevant junctures, whereas PPs modifying NPs are discussed in some more detail in the next chapter. I am, however, not discussing sentence adverbials. The semantic aspect of VP modification by PPs is discussed in Chapters 4 and 5, whereas Chapter 3 is concerned with more syntactic aspects. The following section discusses relevant syntactic evidence, which shows that there is in fact no difference between arguments and adjuncts of this kind with respect to extraction.

3.3. Extraction

Hukari & Levine (1994, 1995) discuss constructions in a number of languages which exhibit particular morpho-syntactic behaviour to mark extraction, or displacement structures ('Unbounded Dependency Constructions', UDC, cf. also Zaenen 1983). They then point out that in these constructions extraction is marked irrespective of whether the displaced element is an argument or an adjunct. I present here the evidence adduced by Hukari & Levine[21].

3.3.1. French Stylistic Inverison

French permits optional subject–verb inversion in the context of extraction structures (discussed i.a. in Kayne & Pollock 1978), a process called 'Stylistic Inversion', and illustrated in (27), (Kayne & Pollock 1978, quoted from Hukari & Levine 1994: 285):

(27) *Où espéraient dîner tes amis?*
where hope.3PL dine your friends
'Where did your friends hope to dine?'

The subject *tes amis* follows the agreeing verbal complex, in contrast to the basic SVO order in French. Stylistic inversion is not triggered by the immediate presence of an interrogative pronoun (as in this case *o ù*), but is sensitive to the displacement context, as can be seen from (28), (Kayne & Pollock 1978, quoted from Hukari & Levine 1994: 285):

(28) *Avec qui croit-elle qu'a soupé Marie?*
with who think-she that-has dine Marie
'Who does she think that Marie has dined with?'

In (28) the subject *Marie* follows the verb *soupé* in the subordinate clause. However, the interrogative pronoun (*avec qui*) is in the matrix clause. Thus, Hukari & Levine conclude that stylistic inversion is licensed in the UDC, that is, between the antecedent and its unification site (i.e. extraction site), and not by the presence of a *wh*-word. The examples in (27) and (28) already show that stylistic inversion is not restricted to arguments, but is also licensed by extracted adjuncts. Further examples provided by Hukari & Levine show the contrast in interpretation between non-inverted and inverted structures (Kayne & Pollock 1978, quoted from Hukari & Levine 1994: 285 (transcription as in the original)):

(29a) *Où/Quand Marie a-t-elle déclaré que Paul était mort?*
Where/When M 3.sg.aux-2.sg.cl.3.sg.fem.cl declared that P was dead
'Where/when did Marie declare that Paul had died?

21 The data are also discussed in Bouma, Malouf & Sag (1997).

(29b) *Où/Quand Marie a-t-elle déclaré que était mort Paul?*
Where/When M 3.sg.aux-2.sg.cl.3.sg.fem.cl declared that was dead P
'Where/when did Marie declare that Paul had died?

The difference between these two sentences is that (29a) has the non-inverted order subject–verb in the lower clause (*... Paul était mort*), while (29b) has the inverted order verb–subject (*... était mort Paul*). The difference in interpretation follows from what has been said so far: (29a) is ambiguous between a reading where the *wh*-phrase is interpreted in the main clause and a reading where it is interpreted in the subordinate clause. (29b), on the other hand is not ambiguous; the *wh*-phrase has to be interpreted in the lower clause, since the lower clause is marked as lying within the UDC. Hukari & Levine conclude (1994: 285/86):

> These examples show quite clearly that French SI [Stylistic Inversion, LM] – a construction which registers UDC paths – is quite sensitive to both argument and adjunct unbounded dependencies constructions, thus providing unequivocal evidence that the latter are bona fide instances of syntactic extraction.

3.3.2. Downstep Suppression in Kikuyu

The Kenyan Bantu language Kikuyu provides further evidence for the identical behaviour of arguments and adjuncts in displacement contexts. The evidence concerns the tonal behaviour of sentence final words in a tensed verb phrase. In sentences without extraction, words like *moanáké*, 'boy', and *káYokó*, 'chicken', retain their final high tone in sentence final position. However, this rule does not apply within a displacement construction (data from Clements et al. 1983, quoted from Hukari & Levine 1995: 212)[22]:

(30) *né káYokó karékó móndo ꜜahéíꜜ ré moànàkè t*
FP chicken which person gave boy
'Which chicken did someone give to the boy?'

(31) *né ré móndo ꜜahéíꜜ ré moanáké káYòkò t*
FP when person gave boy chicken
'When did someone give the boy a chicken?'

In (30), the sentence final *moànàkè* does not retain its high tone since it is within a displacement context – *káYokó* has been extracted. As (32) shows,

22 Transcribed as in Hukari & Levine, but without the distinction between open and close vowels.

káYòkò does not retain its high tone either, that is (31) behaves exactly as (30), although the fronted phrase in (30) is an argument, while the fronted phrase in (31) is an adjunct.

3.3.3. Agreement in Chamorro and Palauan

The next set of data Hukari & Levine present comes from the Austronesian languages Chamorro and Palauan. The relevant reflex of displacement constructions in these two languages is morphological: verbal agreement registers UCD paths. In Chamorro, verbs in displacement structures are marked by a nominalizer as in (32), (from Chung 1982, quoted from Hukari & Levine 1994: 286) and (33), (from Chung & Georgopoulos 1988, quoted from Hukari & Levine 1994: 287)[23]:

(32) *Hafa puno'-mu ni lälu'?*
what? kill+Nmlz-your Obl fly
'What did you kill the fly with?

(33) *Taimänu sagan-ña si Juan __ ?*
how? say+Nmlz-his Unm
'How did Juan say it?

Both (32) and (33) involve non-argument extraction, the former example an 'instrumental', and the second a manner phrase.

In Palauan, displacement structures are marked by a difference in verbal mood. The verb is realis when the displaced phrase is nominative, irrealis for all other displaced phrases. (34a – b), (from Georgopoulos 1985, quoted from Hukari & Levine 1994: 287) show nominative extraction, where the verb in irrealis (34b) is ungrammatical:

(34a) *ng-te'a$_i$ [a kileld-ii a sub __ $_i$]*
CL who R-PF-heat-3s soup
'Who heated up the soup?

(34b) **ng-te'a$_i$ [a le-kileld-ii a sub __ $_i$]*
CL who IR-heat-3s soup

23 All examples in this section are represented as found in Hukari & Levine (1994) except for the use of ' to denote a glottal stop.

Correspondingly, with an extracted object, irrealis mood is required and realis is ungrammatical (35a – b), (from Georgopoulos 1985, quoted from Hukari & Levine 1994: 287):

(35a) *ng-ngera$_i$ [a le-silseb-ii __ $_i$ a se?el-il]*
CL what IR-3 PF-burn-3s friend 3s
'What did his friend burn?'

(35b) **ng-ngera$_i$ [a silseb-ii __ $_i$ a se?el-il]*
CL what R-3 PF-burn-3s friend 3s
Int.: 'What did his friend burn?'

Finally, (36), (from Georgopoulos 1985, quoted from Hukari & Levine 1994: 287) shows that adjuncts behave like objects:

(36) *ng- ker$_i$ [a le- bilsk -au a buk er ngii$_i$ a Ruth]*
CL where IR-3 gave 2s book P it
'Where did Ruth give you the book?

Thus, the data from both Chamorro and Palauan confirm Hukari & Levine's observation.

3.3.4. Irish Complementizers

A further set of relevant data is the distribution of two different forms of the complementizer in Irish, discussed by McCloskey (1979) Hukari & Levine (1995), and Bouma, Malouf & Sag (1997). Irish has two different complementizers, *goN* and *aL*, where the former is found in non-displacement structures, while the latter registers extraction. The following examples illustrate that pattern (from McCloskey 1979, quoted from Hukari & Levine 1995: 205/206)[24]:

(37)

Shîl	*m é*	*goN*	*mbeadh*	*sé*	*ann*
thought	I	COMP	would-be	he	there

'I thought that he would be there.'

(38)

an fear	*aL*	*shîl*	*m é*	*aL*	*bheadh*	_	*ann*
[the man]$_j$	COMP	thought	I	COMP	would-be	e$_j$	there

'the man that I thought would be there'

24 All data in this section are presented as found in Hukari & Levine (1995). In *goN*, N indicates nasal mutation and in *a L*, L indicates lenition.

(39)

an fear	*aL*	*shíl*	_	*goN*	*mbeadh*	*sé*	*ann*
[the man]$_j$	COMP	thought	e$_j$	COMP	would-be	he	there

'the man that thought he would be there'

The examples in (37–39) show the difference between the two Irish complementizers. In (37), *goN* is used, since the sentence does not involve a fronted phrase. In contrast, the example in (38) requires the complementizer *aL*, since *an fear*, 'the man', is extracted (relativized). Note that both complementizers register the extraction path. The sentence in (39) shows that *aL* and *goN* can co-occur, and that *aL* is used up to the extraction site of the fronted phrase, but that after the extraction site, *goN* is used. The following examples show that *aL* is used not only when an argument is extracted (as in (38) and (39)), but also when the fonted phrase is an adjunct (McCloskey 1979, quoted from Hukari & Levine 1995: 206):

(40)

I mBetlehem	*aL*	*dúirt*	*na targaireachtaí*
[in Bethlehem]$_j$	COMP	said	the prophecies

aL	*béarfaí*	*an Slánaitheoir*	_ .
COMP	would-be-born	the Saviour	e$_j$

'It was in Bethlehem that the prophecies said that the Saviour would be born'

(41)

Cén uair	*aL*	*tháing*	*siad*	*n'a bhaile*	_ ?
[which time]$_j$	COMP	came	they	home	e$_j$

The extracted element in (40) is a topicalized locative adjunct, while in (41), a temporal adjunct is questioned. In both cases the choice of the complementizer is *aL*, indicating that there is an extraction path between the adjunct and its eventual position lower in the tree. Thus the distribution of the two forms of the Irish complementizer show that arguments and adjuncts behave alike in extraction contexts. As a last set of data, the next section discusses examples from English.

3.3.5. English

English does not have extraction-sensitive morphological or syntactic alternations. Hence to find evidence for the hypothesis that arguments behave like adjuncts proposed by Hukari & Levine (1994, 1995) is not as straightforward as in the cases considered so far. Hukari & Levine observe that despite the lack of obvious syntactic evidence, arguments and adjuncts behave alike in a way which would be unexpected if they resulted from different underlying structures. The parallel behaviour of arguments and adjuncts can be seen from the fact that both arguments and adjuncts can be extracted out of identical environments, including "... finite and infinite interrogatives, finite and infinite relative clauses (with or without overt *wh*-phrases), topicalizations, clefts, exclamatory wh-constructions, free relatives, and so forth ..." (Hukari & Levine 1994: 289). Some of these parallel structures are illustrated below:

(42a) What did Bill open GAP?
(42b) What did Bill open the door with GAP?

(43a) What does Sally believe Jim opened GAP?
(43b) What bed does Mary believe Sue refuses to sleep in GAP?

(44a) It was a sandwich Steve ordered GAP, not breakfast.
(44b) It was a house we lived in GAP, not a flat.

(45a) It is history I passed GAP, not maths.
(45b) It is Dr Miller I have to finish this for GAP, not Dr Smith.

The examples above illustrate *wh*-questions and clefting, the (a) examples involving arguments extraction, the (b) examples extraction out of adjunct position. Similarly, if extraction of arguments is not possible, neither is extraction of adjuncts:

(46a) *What did Bill, who opened GAP, saw Mary?
(46b) *What did Bill, who opened the door with GAP, saw Mary?

As Hukari & Levine (1994: 290) point out, adjunct extraction is subject to strong and weak cross-over effects in the same way that argument extraction is:

(47a) *?Who$_i$ did Mary claim he$_i$ asked Sarah to visit GAP$_i$?
(47b) *?Who$_i$ did Mary claim he$_i$ asked Sarah to have lunch with GAP$_i$?

(48a) ?Who$_i$ did Mary claim his$_i$ mother asked Sarah to see GAP$_i$?
(48b) ?Who$_i$ did Mary claim his$_i$ mother asked Sarah to have lunch with GAP$_i$?

In the examples above, a co-indexed pronoun intervenes between a fronted phrase and the extraction site, creating a cross-over configuration. Although judgements vary, differences in acceptability result from the position of pronoun, i.e. whether it is embedded in larger constituent or not, but are not affected by the status of the fronted phrase, i.e. whether an argument or an adjunct is extracted. That is to say, cross-over data provide further evidence for the similarity between arguments and adjuncts.

To these distributional facts, the similarity of arguments and adjuncts in quantifier scoping can be added. An indefinite noun phrase in adjunct position can take wide scope over object and subject in the same way that an indefinite object may outscope its subject:

(49a) Every student kissed a local boy.
$\exists x \forall y$ (local_boy(x) & (student(y) -> kiss(y,x)))

(49b) Every student kissed a local boy at a beach.
$\exists x \forall y \exists z$ (beach(x) & (student(y) -> (local_boy(z) & kiss_at(y,z,x)))

Thus indefinite noun phrases can have wide scope both out of argument and out of adjunct position.

The examples in this section thus show that there are a number of contexts in English in which arguments and adjuncts behave alike.

3.4. Summary

The examples introduced in this section show that the distinction between arguments and adjuncts rests mainly on the obligatoriness of the former and the optionality of the latter. However, this distinction does not clearly correlate with any morphological, semantic, or syntactic distinctions. Morphologically, both adverbs and PPs can be obligatory, while semantically, optional PPs may have the same function as arguments. Finally, the syntactic evidence shows that arguments and adjuncts behave alike in languages which overtly indicate that a constituent has been extracted. This evidence taken together indicates that the distinction between arguments and adjuncts is weaker than implied by strict subcategorization.

4. Preliminaries for an Analysis of Verb Phrase Adjunction in LDSNL

The evidence considered in the last section shows that verb phrase adjuncts behave like arguments in several respects. The analysis of arguments and adjuncts should thus reflect their similarity, as well as the difference between the two with respect to optionality. In this section I offer a preliminary discussion of the argument developed in this thesis, namely that verbal subcategorization is underspecified. I begin by showing that adjuncts stand in a daughter relation to the root node, rather than in a LINK relation (4.1.). I then provide a sample derivation of argument extraction, to show how an extracted argument is assigned its place in the eventual tree (4.2.). The derivation is contrasted with an attempted derivation of adjunct extraction, which shows that no obvious place in the tree can be found for the adjunct. I then present three alternative means to expand tree structure and indicate a preferred solution (4.3.). In the final section I discuss the relation between the tree structure needed for adjuncts and verbal subcategorization.

4.1. Adjuncts as Daughters

The data discussed in this chapter show that adjuncts stand in a daughter relation to the root node. As discussed in the last chapter, relative clauses are analysed in LDSNL as LINKed trees; a new tree is built which is LINKed to the head noun. One of the differences between the two structures is related to extraction; LINKed trees are islands for extraction. In LDSNL terms, this means that an unfixed node cannot be fixed in a LINKed tree. Since, as illustrated above, adjuncts and arguments can be extracted under the same circumstances, but neither can be extracted out of relative clauses, adjuncts can not be LINKed. The syntactic evidence discussed above thus indicates that both arguments and adjuncts are in a daughter relation to the root node.

The relevant tree relation can be seen in the tree for a transitive sentence like (50):

(50) Judy closed the window.

The corresponding LDSNL tree (before Completion) is given in (51):

(51) *Tree for* "Judy closed the window"

```
                              • {Tn(0), ?Ty(t)}
                                    / \
• {Tn(00), Fo(judy'), Ty(e)}    • {Tn(01), ?Ty(e -> t)}
                                    / \
   • {Tn(010), Fo(the_window')}  • {Tn(011), Fo(close'),
                    Ty(e)           Ty(e -> (e -> t))}
```

The tree shows that the object *the window* functions as an argument to the verb, which requires two Ty(e) expressions. It is located at Tn(010), that is, in a daughter relation to both the VP node Tn(01) and the root node Tn(0). In the next section I show how an extracted object is assigned to this position in the eventual tree structure, and then develop an analogous derivation for an extracted adjunct, for which, however, no obvious place in the tree can be found.

4.2. Argument Extraction

The object of the example in (50) can be questioned as in (52):

(52) What did Judy close?

As discussed in the preceding chapter, the LDSNL analysis of extraction involves the building of an unfixed node, which is assigned its place in the eventual tree only after further tree structure has been built. I repeat the relevant parse stages here.

At the initial parse stage, the *w h*-pronoun is assigned to an unfixed node, and the root node is annotated with a question feature +Q:

(53a) *Tree for* "What

```
                    • {Tn(0), +Q, ?Ty(t)}

{Tn(0*), Fo(WH), Ty(e)}
```

Ignoring the contribution of the auxiliary verb *did*, the next step is the scanning of the subject and the application of Introduction and Prediction to result in the building of the subject node for *Judy*:

(53b) *Tree for* "What did Judy

```
                    • {Tn(0), +Q, ?Ty(t)}
                   /
 • {Tn(00), Fo(judy'),
            Ty(e)}

 {Tn(0*), Fo(WH), Ty(e)}
```

The next step in the derivation is the application of Introduction and Prediction, which results in the building of Tn(01) with a requirement TODO Ty(e –> t):

(53c) *Tree for* "What did Judy

```
                    • {Tn(0), +Q, ?Ty(t)}
                   / \
 • {Tn(00), Fo(judy'),   • {Tn(01), ?Ty(e -> t)}
            Ty(e)}

 {Tn(0*), Fo(WH), Ty(e)}
```

However, the type information from the verb *close*, i.e. Ty(e –> (e –> t)), does not match the requirement holding at Tn(01). The lexical information from *close* licenses in this situation the building of two new nodes, in accordance with the verb's subcategorization requirement:

(54) *Lexical Entry for* close

IF	? Ty(e –> t)
THEN	put(?$<d_0>$ Ty(e))
	make($<d_1>$), put(Fo(close'), Ty(e –> (e –> t)))
ELSE	abort

The lexical information from *close* drives the process of tree building and ensures that its subcategorization requirements are projected into the tree by building a functor node for its formula and type value, and by building an argument node with the requirement TODO Ty(e). Thus the following transition is licensed:

(53d) *Tree for* "What did Judy close

• {Tn(0), +Q, ?Ty(t)}
/ \
• {Tn(00), Fo(judy'), Ty(e)} • {Tn(01), ?Ty(e –> t)}
/ \
{Tn(010), ?Ty(e)} • {Tn(011), Fo(close'), Ty(e –> (e –> t))}

{Tn(0*), Fo(WH), Ty(e)}

At this stage, the unfixed node has not yet been assigned its eventual location, but it is clear how this can be done. The *w h*-pronoun is of Ty(e), as indicated at Tn(0*). Furthermore, the DU has to be fixed at a position in the tree which is lower than Tn(0), that is, at a daughter node. These two specifications match exactly the information holding at Tn(010). It is a daughter node of the root node, and there is a requirement TODO Ty(e). By merging Tn(010) with the underspecified node Tn(0*), the tree can be completed with no requirement outstanding, and all nodes being fixed:

(53e) *Tree for* "What did Judy close?"

• {Tn(0), +Q, ?Ty(t)}
/ \
• {Tn(00), Fo(judy'), Ty(e)} • {Tn(01), ?Ty(e –> t)}
/ \
• {Tn(010), Fo(WH), Ty(e)} • {Tn(011), Fo(close'), Ty(e –> (e –> t))}

The *w h*-pronoun is found in the eventual tree in object position as desired, and the tree is structurally identical to the tree for the corresponding declarative sentence in (51). Crucially, the unfixed node in (53d) could be assigned a position at a daughter node – otherwise, the derivation would not have been completed. The presence of the daughter node Tn(010) results from lexical instructions from the verb, so it derives ultimately from subcategorization information. The status of the questioned constituent as an argument to the verb is thus crucial to ensure its eventual place in the tree.

4.3. Adjunct Extraction

Given the analysis of argument extraction discussed in the preceding section, and in view of the fact that arguments and adjuncts behave alike with respect to extraction, a derivation for adjunct extraction is attempted in this section following the steps described in the argument extraction derivation above. This leads to a discussion of how an LDSNL analysis of verb phrase adjunction can be developed, a discussion which will form the background for the analysis proposed in Chapter 3.

As an example, consider the adjunct in (55a) and the corresponding question in (55b):

(55a) Judy was sleeping in a sleeping bag.

(55b) What was Judy sleeping in?

The initial steps of an LDSNL derivation for (55b) are the same as those for argument extraction. First, the *wh*-pronoun is assigned to an unfixed node, and the root node is annotated with +Q. Since I do not analyse tense here, the second step is – as above – the introduction of the subject and the building of the VP node by Introduction and Prediction:

(56a) *Tree for* "What was Judy

```
                         • {Tn(0), +Q, ?Ty(t)}
                        / \
• {Tn(00), Fo(judy'),        • {Tn(01), ?Ty(e -> t)}
          Ty(e)}

{Tn(0*), Fo(WH), Ty(e)}
```

This is the state of the derivation when *sleep* is scanned. I assume that *sleep* is intransitive, as stated in the following lexical entry:

(57) *Lexical Entry for* sleep

```
IF      ? Ty(e -> t)
THEN    put(Fo(sleep'), Ty(e -> t))
ELSE    abort
```

The entry states that *sleep* can be introduced at a VP node, and that it annotates the node with its formula and type value, thereby fulfilling the requirement obtaining at that node:

(56b) *Tree for* "What was Judy sleeping

• {Tn(0), +Q, ?Ty(t)}
/ \
• {Tn(00), Fo(judy'), Ty(e)} • {Tn(01), Fo(sleep'), Ty(e –> t) ◊}

{Tn(0*), Fo(WH), Ty(e)}

Note that the pointer at this stage is still at Tn(01), so that Completion can apply to record the fulfilled requirement as a modal statement at the root node Tn(0), where Elimination applies. With respect to subcategorized predicate–argument structure, the tree in (56b) is completed, since the combination of Fo(judy') and Fo(sleep') would result in a proposition of Ty(t). Yet the derivation cannot be completed, since there is further input, namely the preposition *in*, and there is still an unfixed node, which needs to be fixed for the derivation to be successful. At this state, then, the parallelism between the derivation of argument extraction and the derivation of adjunct extraction ends. Whereas for the argument, the unfixed node could be located in the tree at a daughter node resulting from the verb's subcategorization information, there is no obvious place in the tree in (56b) for the adjunct at the unfixed node. As it stands, the tree is unfinished and the derivation is aborted.

What can be done? There is still the preposition, which might lexically contribute to a resolution of the problem, for example, by building a new Ty(e) node. Notwithstanding the LINK operation, there are three general possibilites to expand a tree like the one in (56b), and I sketch them here so as to prepare for the discussion in the following chapter.

The first possibility is inappropriate for the example discussed here, but should briefly be mentioned. This is to continue the tree after Completion has applied and the pointer has moved to the root node:

(58)

• {Tn(0), Fo(β(α)), Ty(t) ◊}
/ \
• {Tn(00), Fo(α), Ty(e)} • {Tn(01), Fo(β), Ty(e –> t)}

In a situation like this, the tree could be expanded upwards by building a new root node (a 'grandmother' node):

(59)

```
                      • {?Ty(t)}
                     /\
   • {?Ty(t -> t}   • {Tn(0), Fo(β(α)), Ty(t)}
                       / \
 • {Tn(00), Fo(α),      • {Tn(01), Fo(β), Ty(e -> t)}
          Ty(e)}
```

I have omitted several details in the tree in (59), such as the value of the tree-node predicate, or how a Ty(e) expression could be integrated into the tree. However, for the example discussed here, this tree continuation is not possible, since the new nodes are not in a daughter relation to the original root node. Intuitively, the continuation shown in (59) could be employed for the analysis of sentence adverbials such as *possibly*, but it is not a viable option for adjuncts such as those discussed here. I will thus not explore this possibility further.

Another possibility to integrate adjuncts into a tree like in (56b) might be to analyse the adjunct as operating on the VP node. Thus, a possible eventual tree for the example in (56b) might be the following:

(60) *Possible Tree for* "What was Judy sleeping in?"

```
                                  • {Tn(0), +Q, ?Ty(t)}
                                 / \
 • {Tn(00), Fo(judy'), Ty(e)}    • {?Ty(e -> t)}
                                   /\
     • {Fo(sleep'), Ty(e -> t)}   • {?Ty((e -> t) -> (e -> t))}
                                     /\
             {Fo(WH), Ty(e)}   • {Fo(in),
                                  Ty(e -> ((e -> t) -> (e -> t)))}
```

The continuation in (60) is downwards, and a new Ty(e) node is built which does stand in a daughter relation to the root node. The new functor node is built by the preposition, which acts as the main functor in the tree. Note that, as in the tree above, I have left out the tree addresses of the relevant nodes in (60). Although here it is clear what they should be, it is not clear how they are assigned; in particular the tree-node value for the DU introduced by the verb would have to be assigned a new address. In the next chapter, I argue that a continuation like the one in (60) is impossible in LDSNL, so that it is not a viable analysis for verb phrase adjunction.

The third alternative to be considered here is the one which I argue in the next chapter to be the best solution, namely to analyse verb phrase adjuncts as optional arguments of verbs[25]. A corresponding tree is shown below:

(61) *Possible Tree for* "What was Judy sleeping in?"

```
                                  • {Tn(0), +Q, ?Ty(t)}
                                 / \
• {Tn(00), Fo(judy'), Ty(e)}   • {Tn(01), ?Ty(e -> t)}
                                   / \
          • {Tn(010), Fo(WH),   • {Tn(011), Fo(sleep'),
                       Ty(e)}        Ty(e -> (e -> t))}
```

In (61), the verb appears as transitive, and a corresponding new argument node Tn(010) has been built at which the unfixed node has been located. The new node is a daughter of the root node and is below the original VP node, so that under this analysis adjuncts are transparently analysed as arguments of the verb. The syntactic contribution of the preposition can then be seen as licensing the necessary process of tree continuation. However, as pointed out in this chapter, adjuncts are not oligatory constituents of the verb phrase, they are not subcategorized for by the verb. One of the points discussed in detail in the following chapters is concerned with the role of subcategorization in view of an analysis of verb phrase adjunction which treats adjuncts as optional arguments. A preliminary discussion is offered in the next section.

4.4. Adjuncts and Subcategorization

A putative solution to the problem posed by the adjunct extraction derivation of the last section is to simply assume that there are two lexical entries for *sleep*, one intransitive one, and one which allows for the introduction of an argument. The derivation discussed above would then simply involve the right choice of predicate, for example in the case of *sleep*, discussed above, the one which allows for the introduction of an additional Ty(e) expression. The difference between arguments and adjuncts would under this view result from different lexical entries for verbs. This would explain the optionality of adjuncts since the choice of a particular lexical entry is in general optional.

However, while a lexical solution is an appropriate analysis for optional arguments of verbs with both an intransitive and a transitive use, such as, for

25 The alternatives illustrated here have been discussed in the literature. In Chapter 4, the analyses of Dowty (1979), McConnell-Ginet (1982) and Copestake, Flickinger & Sag (1997) are discussed in detail.

example, *drink*, or *read*, it is implausible for verb phrase adjunction in general, under the assumption that adjuncts are analysed as (non-lexical) optional arguments, as implied in the tree continuation in (61), above. Under this view, all verbs allow for the optional introduction of Ty(e) expressions, as seen in (62):

(62) Mary was singing her favourite song at the top of her voice with her sister for their parents at Christmas in the drawing room.

In view of the generally free possibility of introducing Ty(e) expressions, which I take here to include PPs, a point discussed in detail in the next chapter, lexical subcategorization statements appear to be the wrong tool. Rather, a verb such as *sing* may be lexically categorized as requiring at least one Ty(e) expression, or, in its transitive use, two Ty(e) expressions, but, given the view of adjunction sketched here, there should in addition be the possibility to express that *sing* on occasion may combine with seven Ty(e) expressions, as in the example in (62). Thus, possible type values of *sing* include:

(63) *Possible Type Values of* sing

Ty(e –> t)
Ty(e –> (e –> t)
Ty(e –> (e –> (e –> (e –> (e –> (e –> (e –> t)))))))

The type values in (63) illustrate the intransitive and the transitive use of *sing*, as well as the necessary type value for the analysis of the example in (62) under the assumption that the Ty(e) expressions are introduced as arguments of the verb. While the first two type specifications might be regarded as lexical information from *sing*, the last type is the result of the general process of adjunction. In the next chapter I propose an analysis which models this general process of adjunction as type underspecification, that is to say, as an element of variability in verbal subcategorization. The remainder of the thesis is then mainly concerned with the formalization and implications of this view.

5. Conclusion

In this chapter I have more precisely stated what the problem discussed in this thesis is and determined the empirical range envisaged. I have introduced the relevant theoretical notions of verb phrase and subcategorization, and how these notions are defined in LDSNL. I have pointed out that the notion of strict subcategorization implies a distinction between obligatory and optional constituents of the verb phrase, i.e. between arguments and adjuncts, but that this distinction has no morphological, semantic, or syntactic correlates. In order to demonstrate this latter point a number of empirical data have been introduced and discussed. The conclusion to be drawn from the evidence presented in this chapter is that arguments and adjuncts differ from each other mainly with respect to optionality, so that adjuncts, once they are introduced, should be analysed as arguments. I have then turned to the question of how this observation can be expressed in LDSNL and have presented two sample derivations, contrasting argument and adjunct extraction. I have presented three possibilities for expanding tree structure so as to incorporate adjuncts, and the two possibilities relevant for VP adjuncts will be discussed further in the following chapter. However, I have already indicated that I propose to analyse adjuncts as optional arguments of the verb, and that this analysis leads to the introduction of underspecification into the type information provided by verbs.

Chapter 3

*Formalizing Verbal Underspecification: e**

1. Introduction

In the last chapter, I have presented evidence to support the view that the distinction between arguments and adjuncts mainly reflects the difference between obligatory and optional presence in the verb phrase, but that adjuncts behave syntactically like arguments once they are introduced. In this chapter I propose various ways in which these facts can be modelled in the LDSNL model introduced in Chapter 1. In particular, I show that basic LDSNL assumptions about incrementality and underspecification lead naturally to the view that verbal subcategorization can be analysed as underspecified. I review these assumptions in Section 2. In Section 3, I discuss a possible analysis which assumes that adjuncts, rather than verbs are underspecified, and show some problems with this view. In Section 4, I develop an analysis of underspecified type values for verbs, and show how this type interacts with the overall LDSNL system. The results of this chapter are summarized and discussed in Section 5. The discussion in this chapter is concerned with the dynamics of tree building, while the semantic interpretation of adjuncts in general, and of the specific analysis of verbal underspecification modelled in this chapter in particular will be discussed in the next two chapters.

2. Two Basic LDSNL Assumptions

As has been pointed out in Chapter 1, LDSNL models the process of utterance interpretation which the hearer is required to perform in order to derive contextual effects. This process includes the combination of lexical building blocks (supplied by phonological accessing) into larger structural units. The process is strictly left-to-right in the sense that each lexical building block is taken in turn and, by application of relevant rules, integrated into the unfolding tree structure. More formally, the structure building process is *incremental*. However, the mapping from 'surface structure', i.e. here the linear order of incoming lexical items, to conceptual structure, the output of the parse, is not one-to-one. Incrementality is achieved by allowing *underspecification* to be part of natural language expressions. These two assumptions are discussed in more detail in this section, since they are part of the motivation for the analysis of verbal underspecification proposed in Section 4.

2.1. Incrementality

To say that the process of building interpretable structures is incremental reflects the observation that, on the one hand, hearers receive syntactic information in units, as accessed in phonological domains, one block after the other, and that, on the other hand, syntactic processing is fast and automatic. The model should thus reflect that interpretations are built by knowing increasingly more about the eventual tree. I express this aspect of incrementality as 'informational incrementality' as in (1):

(1) *Incrementality (informational)*

Interpretation always involves an increase in information.

The aspect of incrementality expressed in (1) involves some idealization vis-a-vis human parsing as reported in the literature (e.g. Frazier & Fodor 1978, Berwick & Weinberg 1984, Fodor 1995, Gorell 1995), where there is agreement that the human parser allows for local restructuring of already established structure, for example in certain types of garden path utterances. This means that (1) is too strong as a statement of real time human parsing. It should be borne in mind, however, that the LDSNL model is a model of competence in the sense discussed in Chapter 1, which by claim is amenable to psycholinguistic evidence, but that it is not a proper model of a parser. There are a number of avenues to incorporate the restructuring evidence into the model, for example by allowing for several parallel parses, or by relaxing the requirement of incrementality in certain respects. In the present context, however, it seems justified to maintain (1) as it stands, since it leads to the formulation of more precise syntactic and semantic characterizations of incrementality, as presented anon, and since it is warranted as a theoretical assumption by the overall cognitive argument presented in Chapter 1.

The strong version of informational incrementality in (1) is paralleled in the syntax of LDSNL by the syntactic aspect of incrementality expressed here as (2):

(2) *Incrementality (syntactic)*

> The process of building tree structure is defined as an incremental process. No process is defined which removes previously built structure.

This aspect of incrementality is the most important one in the following discussion of underspecified type values in the verb phrase. This syntactic restriction constrains the LDSNL system considerably – both information from the lexicon and the transition rules have to be formulated such that they guarantee that all syntactic information is exploited at every step in a derivation, while at the same ensuring that there is no 'back-tracking', that is, that every step in a derivation results either in leaving the tree as it is, or as developing it, but that no step in the derivation can ever remove nodes, or node descriptions from a given tree. The consequences of this restriction for the analysis of adjuncts will be discussed below.

Finally, the corresponding semantic notion of incrementality can be expressed as a requirement on tree structure:

(3) *Incrementality (semantic)*

> Tree structure once built cannot be undone.

The notion of semantics invoked here is the structural notion of the semantics of tree structure – the 'operational' semantics of the LDSNL system. The model-theoretic semantic interpretation of the natural language string is defined over the output of the structure building process, in particular over the information from the formula values. For the purposes here, it is the first, structural notion of semantics which is important.

Given the three aspects of incrementality discussed so far, the following 'principle of incrementality' in LDSNL can be stated:

(4) *Incrementality*

> The process of utterance interpretation as modelled in LDSNL is informationally, syntactically and semantically incremental.

Since the process of utterance interpretation in LDSNL includes information provided in the lexicon as part of the tree building process, the following two corollaries result from (4):

(5) *Corollary 1: The LDSNL Projection Principle*

Information which is established in the lexicon remains constant (unchanged) in a derivation.

(6) *Corollary 2: Lexical Projection of Type Values*

Type values established in the lexicon remain constant (unchanged) in a derivation.

That is, since lexical information is part of the derivation, it is subject to the principle of incrementality. Lexical information cannot be changed during a derivation (5). This includes in particular any type values established in the lexicon (6). If some lexical information includes the predicate Ty(e), it has to be associated with a node with a requirement of Ty(e), from which it follows that, as discussed in the preceding chapter, type shifting or function composition over types are not recognized processes in LDSNL, in contrast to Categorial Grammar.

The strong commitment to incrementality expressed in the principle of incrementality in (4) has the immediate consequence that the question, *which information is available at a given step in a derivation?* is of central importance to the LDSNL enterprise. The answer to the question from an LDSNL perspective has predominantly to do with *the underspecification of natural language.*

2.2. Underspecification

Certain instances of underspecification in natural language are widely regarded as such – pronominal expressions, for example, do not fully encode their referential information; they are underspecified in regard to their (model-theoretic) semantic contribution. In order to know what a pronominal expression refers to, it has to be pragmatically enriched, exploiting contextual and background knowledge. In LDSNL, this is modelled as an underspecified formula value, such that, for example, a personal pronoun encodes a metavariable, possibly with certain restrictions, as its formula value, which is taken to be an instruction to the hearer to establish some suitable referent. While this treatment seems to be rather uncontentious for the semantic interpretation of at least some pronouns[26], it is not usually taken to be related to syntactic structure.

In LDSNL, however, structural, syntactic underspecification is taken to be a key feature of natural language, providing the basis for the analysis of a range of syntactic phenomena, including topic structures and question formation, where there is a mismatch between the surface position of a given constituent in a natural language string and its eventual position in the semantic tree. As demonstrated in the preceding chapters, fronted constituents are modelled as having an initially underspecified tree location, which is only resolved at some later stage in the derivation. That is, question word 'movement' is in LDSNL terms an instance of syntactic underspecification. It is to a large extent due to the possibility of having structural underspecification that the strong requirement of incrementality discussed in the preceding section can be maintained; a displaced constituent is not assigned an initial or intermediate place in the tree, but rather encodes the lack of any definite information as to its location. In that way, tree structure is not established, and resolving the final position of the displaced constituent can clearly be seen as an increase in information. The relevant steps of the derivation with unfixed constituents discussed in the last chapter demonstrate this:

(7a) *Tree for* "What

• {Tn(0), +Q, ?Ty(t)}

{Tn(0*), Fo(WH), Ty(e)}

The DU introduced by *what* is not integrated into the tree, since this is not warranted by the available information at this stage in the derivation. The syntactic information encoded by this DU is underspecified with respect to its eventual location in the tree. During the following transitions, the information that the DU is an as yet unfixed part of the tree is available at every step of the derivation. The DU is eventually assigned a fixed position in the tree when the underspecification is resolved:

26 The analogy with the interpretation of pronouns depends on the particular analysis of anaphoric expressions, some of which can be analysed, for example, as bound variables, rather than instances of underspecification. The LDSNL view is discussed in more detail in Chapter 5 in connection with the more general argument for mental representations.

(7b) *Tree for* "What did Judy close ?"

```
                              • {Tn(0), +Q, ?Ty(t)}
                                        / \
• {Tn(00), Fo(judy'), Ty(e)}    • {Tn(01), ?Ty(e -> t)}
                                        / \
        • {Tn(010), Fo(WH), Ty(e)}   • {Tn(011), Fo(close'),
                                        Ty(e -> (e -> t))}
```

In the final stage of the derivation, the underspecification of Tn(0*) is resolved as Tn(010) by an application of Merge, and the DU is integrated into the tree. With regard to the present discussion, it is important that the resolution of Tn(0*) to Tn(010) is an increase in information, that no previously established tree structure has been changed, and that all semantic relations holding before the final step in the derivation still hold after that step. In other words, underspecification and the resolution of underspecification can be expressed within a derivation which is incremental in the sense defined in (4).

The discussion so far has highlighted the relation between incrementality and underspecification. While returning to incrementality presently, a final observation is directed at the forms of underspecification introduced so far. As pointed out already, LDSNL employs underspecified values of the Formula predicate, for example Fo(WH) for an interrogative pronoun as English *who*, or Fo($U_{speaker}$) for a personal pronoun as English *I*. Additionally, there are underspecified values for the Tree Node predicate, as for example Tn(0*) for a DU holding somewhere below the top node Tn(0), where '*' is a short-hand notation for the Kleene star operation over the daughter modality <d>. This leads to the question of whether there are, analogously to the Formula and Tree Node predicates, underspecified values for the Type predicate. In other words, since underspecification is taken in LDSNL to be a fundamental structural characteristic of natural language, expressed as underspecified values of structural predicates, it is a natural extension of the system to inquire into the underspecification of Type values, which is indeed what I propose to do in the remaining sections of this chapter.

In summary, the discussion so far shows that a successful LDSNL treatment of adjunction, to which I turn presently, has to provide an analysis which allows only incremental derivations, without tree restructuring or changing lexical information. In doing so, it might be helpful, but not necessary, to take advantage of the relation between incrementality and underspecification. There is, however, theory internal independent motivation for exploring the possibility of underspecification of values for the

Type predicate, building on an analogy with Formula and Tree Node predicates.

3. Adjunction Rules

The first possible analysis of adjunction I discuss is a development of one of the two alternative proposals sketched at the end of the last chapter and follows the Adjunction rule in Kempson, Meyer-Viol & Gabbay (1999)[27]. Variations of this rule are discussed, including a version which assumes that the basic clausal relationship between a verb and subcategorized complements is fully specified, and that adjuncts can freely be added at several nodes of a tree. This is achieved by assigning an unspecified Type value to prepositions, so that prepositional adjuncts can be added. However, this approach is problematic, since it implies that tree structure has to be undone.

3.1. Adjuncts as Funtors

As was discussed in the last chapter, there are two possible ways to analyse verb phrase adjuncts in LDSNL, namely an analysis which treats adjuncts as functors, and one which treats adjuncts as arguments to verbs. In this section, I discuss a putative analysis which treats adjuncts as functors, while in the second half of this chapter, I propose an analysis which treats adjuncts as arguments.

To begin with, consider the example with a lexical adverb in (8):

(8) Mary smiled nervously.

In (8), *nervously* might be analysed as acting as a modifier of the verb, which is of Ty(e -> t). A possible derivation for the sentence might then proceed as follows:

27 The Adjunction rule discussed here was included in the draft version of Kempson, Meyer-Viol & Gabbay (1999) at the time of writing, but has subsequently been refined. The new version employs a Kleene star specification similar to the formulation I develop in this chapter, but crucially retains the underspecification of the adjunct, as opposed to the predicate. The overall motivation for my proposal is thus not affected by this development, and I keep the section as it is.

(9a) *Tree for* "Mary smiled

```
                     • {Tn(0) ? Ty(t)}
                          / \
• {Tn(00), Fo(mary'),          • {Tn(01), Fo(smile'), Ty(e -> t) ◊}
   Ty(e)}
```

The stage of the derivation displayed in (9a) is reached after the information from *smile* is incorporated into the tree (ignoring, again, tense). The derivation cannot be completed here, since there is more lexical input, namely from *nervously*. On the assumption that *nervously* acts as a modifier, it is lexically of Ty((e -> t) -> (e -> t)). In order to incorporate the adverb into the tree, the DU holding at Tn(01) has to be relocated so as to act as the argument to the modifier:

(9b) *Tree for* "Mary smiled nervously

```
                          • {Tn(0), ?Ty(t)}
                              / \
• {Tn(00), Fo(mary'),     • {Tn(01), ?Ty(e -> t)}
   Ty(e)}                        / \
                   • {Tn(010),       • {Tn(011),
                   Fo(smile'),         Fo(nervously'),
                   Ty(e -> t)}         Ty((e -> t) -> (e -> t))}
```

In (9b), the DU previously holding at Tn(01) is 'lowered' to Tn(010), a new node is created for the DU introduced by *nervously*, namely Tn(011), and a second new node is created, Tn(01), with a requirement of Ty(e -> t), which can be satisfied by Completion. I will argue in the next section that this process of lowering is problematic, but assume for the moment that it is a possibility.

Under this analysis, then, the adverb is lexically of Ty((e -> t) -> (e -> t)), that is a verb phrase modifier. However, the analysis could be extended to include examples such as those in (10):

(10a) Mary looked out of the window nervously.
(10b) Mary looked nervously out of the window.
(10c) Nervously, Mary looked out of the window.

In (10), *nervously* seems to act as a modifier at different levels of clause structure, probably as a verb phrase adverb in (10a), as verb adverb in (10b), and as sentence adverb in (10c)[28]. The analysis described here could be extended to these cases by assuming that adverbs are lexically associated with a schematic type value. That is, an adverb such as *nervously* does not lexically encode a type value Ty((e –> t) –> (e –> t)) as assumed above, but rather, its type value is a variable, which is instantiated by the type value of the relevant node of attachment:

(11) *Lexical Type for Adverbs*

Ty(X –> X), where X = any Type value.

Possible Instantiations

Sentence Adverb:	Ty(t –> t)
VP Adverb:	Ty((e –> t) –> (e –> t))
IntransV Adverb:	Ty((e –> t) –> (e –> t))
TransV Adverb:	Ty((e –> (e –> t)) –> (e –> (e –> t)))
DitransV Adverb:	Ty(((e –> (e –> (e –> t))) –> (e –> (e –> (e –> t))))

By assuming the schematic value in (11), the adverb can be introduced at any node in the tree, while adverbs can still be characterized as being lexically unambiguously of a single underspecified type. Note, however, that VP and intransitive verb adverbs are of the same type. The particular concept of type underspecification assumed in (11) is that adverbs have no type specification in the lexicon at all, and that their type value is specified according to the expression they modify. That is, type underspecification is here modelled as a variable over standard types.

On the analogy with adverbs, PPs in the verb phrase might be analysed as underspecified adverbials. This is the assumption underlying the possible tree for adjunct extraction discussed in the last chapter, repeated below:

(12) *Possible Tree for* "What was Judy sleeping in?"

```
                          • {Tn(0), +Q, ?Ty(t)}
                               / \
• {Tn(00), Fo(judy'), Ty(e)}      • {?Ty(e -> t)}
                                      /\
        • {Fo(sleep'), Ty(e -> t)}      • {?Ty((e -> t) -> (e -> t))}
                                            /\
                  • {Fo(WH), Ty(e)}   • {Fo(in),
                                         Ty(e -> ((e -> t) -> (e -> t)))}
```

28 I do not discuss these differences in detail since I drop (proper) adverbs presently.

In (12), the preposition builds an argument node for the Ty(e) expression and combines with it to result in a verb phrase modifier. The preposition in (12) might thus be analysed as being lexically of the type below:

(13) *Putative Type for Prepositions*

Ty(e -> (X -> X))

Under this analysis, a preposition combines with a Ty(e) expression to result in an adverb. The underspecification can be resolved according to the node to be modified.

In summary, an analysis of adjuncts as functors can be formulated by assuming that an operation of tree stretching, or lowering is possible, and by assuming that prepositions combine with a Ty(e) expression to result in an underspecified adverbial. In the following section I show that both these assumptions are problematic.

3.2. Adjunction Rule

The analysis of adjuncts sketched in the preceding section can be formally stated by formulating an Adjunction rule which licenses an operation of 'sub-tree lowering' in order to introduce a new node in the middle of a tree. The rule is illustrated with abstract trees in (14):

(14a) *Adjunction: Input Tree*

```
                         • {Tn(n)}
                        / \
            {Tn(n0)}•      • {Tn(n1), Ty(Y)}
                          / \
{Tn(n10), Ty(X -> Y)} •      • {Tn(n11), Ty(X)}
```

The tree in (14a) is any given (sub-)tree, where 'n' may be any sequence of 0 and 1. Similarly, the variables X and Y may stand for any well-formed Type value. The Adjunction rule now licenses the introduction of a new node with a DU of Type Ty(Y -> Y), that is applying to the DU with Ty(Y) at Tn(n1) in (14a). The new node thus is a sister to Tn(n1) and results, furthermore, in another new node of Ty(Y) above the old Tn(n1) and the new sister node:

(14b) *Adjunction: Output Tree*

```
                         • {Tn(n)}
                        /\
              {Tn(n0)}•    • {Tn(n1), Ty(Y)}
                           /\
        {Tn(n10), Ty(Y)} •    • {Tn(n11), Ty(Y -> Y)}
                         /\
{Tn(n100), Ty(X -> Y)} •    • {Tn(n101), Ty(X)}
```

The new nodes introduced by the Adjunction rule are indicated in bold print; they are (the new) Tn(n1) and (the new) Tn(n11). As a result of this 'tree-stretching' operation, the old nodes Tn(n1), Tn(n10) and Tn(n11) now become Tn(n10), Tn(n100) and Tn(n101) respectively. The Adjunction rule thus licenses the introduction of new nodes 'into' an already established tree[29]. However, the rule is problematic with respect to incrementality.

3.3. Adjunction: Incrementality

On the surface, the Adjunction rule does not seem to respect incrementality, since it involves the restructuring of already established structure; already established Tree Node values are changed. However, it might be objected that the relative tree relations between the nodes already established remain unchanged. This means that, irrespective of the actual Tree Node values, the relation between nodes as expressed in the tree node modalities <d> and <u> remain unchanged. For example, from the position in the tree where the DU with Ty(X) holds, it was and is true that the statement '<u> Ty(Y)' ("above of me is a DU with Ty(Y)") is true, as is '<u><d> Ty(X -> Y)' ("at my sister Ty(X -> Y) holds"). This is so, because all of these nodes have been lowered uniformly, namely 'one down'. Similarly, from the position of Tn(n), it was and is true that '<d> ...' (meaning here "down of me holds whatever it is that holds down of me"). This is true for two different reasons; for the left daughter, because both nodes have not been changed, and for the right daughter, because by definition the Adjunction rule introduces an identical node to the node the adjunct takes as an argument, i.e. since the adjunct is of type Ty(Y -> Y), the Type value holding at the sister node is identical to the Type value holding at the mother node. For the nodes considered so far, it is thus indeed true that the relative tree positions hold. However, for the original node with Ty(Y), it is not

29 This analysis of adjunction follows the categorial grammar analysis found e.g. in Montague (1973), Dowty (1979), and Morrill (1994). Dowty's analysis is discussed more fully in the next chapter.

true; before adjunction, both '<u> ...' (here, "above me holds whatever holds above me") and '<u><d> ...' (here, "at my sister holds whatever holds at my sister") were true, whereas after adjunction '<u> Ty(Y)' ("above of me is a DU with Ty(Y)") '<u><d> Ty(Y –> Y)' ("at my sister Ty(Y –> Y) holds") are true. One might argue that this particular difference is warranted, since it is after all the point of the rule to introduce these two DUs. But there is another problem. It is in general not true that relative tree relations are unchanged once the modality operators are iterated. For example, before Adjunction it was true for the node where Ty(X) holds that '<u><u> ...' (here, "up of up of me holds whatever up of up of me holds"), while after Adjunction the correct statement is '<u><u> Ty(Y)'[30]. Similarly, from Tn(n), before Adjunction '<d><d> Ty(X)' is true, whereas after Adjunction '<d><d> Ty(Y)' and <d><d> Ty(Y –> Y), as well as '<d><d><d> Ty(X)', but not '<d><d> Ty(X)' hold. Iteration of operators, however, cannot be excluded, since it is needed for the characterization of <d>*, that is, for the characterization of underspecified tree locations, and since iterated modality statements are part of lexical instructions. It might be possible to introduce revisions into the LDSNL system, so that no reference to iterated modality operators is necessary, other than <d>*, by restricting lexical specifications[31]. This would mean that the tree restructuring involves only irrelevant information, and thus that the Adjunction rule does respect incrementality in the sense that relative, but not absolute, tree node locations count as established tree structure, and provided that relative tree locations are sufficiently described by modality statements with one modality operator only. The Adjunction rule discussed here, then, does provide a possible analysis of verb phrase adjunction, but it implies non-trivial revisions of the overall LDSNL system and it does not, as it stands, respect incrementality in a clear sense. Thus, with respect to both conceptual assumptions and formal tools, the Adjunction rule does not present an optimal analysis of verb phrase adjunction in LDSNL.

3.4. Adjunction: Type Values

In addition to the problems pointed out in the last section, there is the further problem that the Adjunction rule is not the right analysis for the adjuncts

30 From now on I omit the English glosses.

31 A possible way to do this is to use the go predicate and specify pointer movement not in relation to tree nodes, e.g. go<u_1>, but in relation to the intended endpoint of the movement, e.g. go(<u*> ?Ty(X)), i.e. "go up to a node where Ty(X) holds". I am not sure, however, if such an analysis could fully replace iterated modality operators.

discussed in the last chapter. It results in problems with respect to the typing of prepositions, and with respect to the semantic structure assigned to PPs modifying the verb.

Under the Adjunction rule, PPs act as functors which take another functor as argument. For example, the relevant type information for the preposition for the tree in (12) is shown below:

(15) {Fo(in), Ty(e -> ((e -> t) -> (e -> t))))}

With the typing in (15) the preposition combines with a noun phrase (which is of Ty(e)) to give the necessary type Ty((e -> t) -> (e -> t)), which combines with an expression of Ty(e -> t) to result in another expression of Ty(e -> t), as required by the Adjunction rule. However, this analysis implies that PPs always act as modifiers, and is thus problematic for subcategorized PPs[32]:

(16a) Jane put the book on the shelf.
(16b) *Jane put the book
(16c) *Jane put the book the shelf

The problem here is that, as discussed in the preceding chapter, the distinction between PPs and NPs is not co-extensive with the distinction between arguments and adjuncts, so that any attempt to derive the latter distinction from lexical typing of the former encounters problems with these cases.

A second point to note about the Adjunction rule is the formula value which it derives at the Ty(t) node. Assuming that the typing problems mentioned above can be solved, a derivation with the Adjunction rule of an example like (17) looks as follows:

(17) John baked a cake for Mary.

32 The same argument holds for subcategorized adverbs, illustrated in the last chapter.

(18) *Tree for* "John baked a cake for Mary"

```
              • {Tn(0), Fo((for'(mary'))((bake'(cake'))(john')), Ty(t)}
                                   / \
      • {Tn(00), Fo(john'),     • {Tn(01), Fo((for'(mary'))((bake'(cake'))),
                  Ty(e)}           Ty(e -> t)}
                                             / \
         • {Tn(010), Fo((bake'(cake')),     • {Tn(011), Fo(for'(mary')),
               Ty(e -> t)}                      Ty((e -> t) -> (e -> t))}
                    /\                                   /\
      • {Tn(0100),   • {Tn(0101),          • {Tn(0110),   • {Tn(0111),
      Fo(cake'),      Fo(bake'),           Fo(mary'),      Fo(for'),
      Ty(e)}          Ty(e -> (e -> t))}   Ty(e)}          Ty(e -> (X -> X)}
```

In the resulting structure in this derivation, the position of the adverb node Tn(011) is such that the adjunct acts as functor which takes the verb phrase at Tn(010) as argument. However, as noted in the last chapter, some adjuncts behave as arguments to the verb, rather than as functors taking the verb, or the verb phrase, as arguments, so that for those adjuncts the adjunction rule gives the wrong analysis. The semantic aspects of this distinction are discussed in the next chapter, but here it is worth noting that locating the underspecification of type values, as implied in the Adjunction rule, in the adjunct rather than in the predicate will always result in the adjunct acting as functor. In the next section, I develop an analysis which locates the underspecification involved in adjunction in the verb, and which introduces adjuncts as optional arguments to the verb.

4. Underspecified Verbs as e*

In this section I come to the main proposal of this thesis. I propose a formalization of verbal subcategorization which treats verb phrase adjuncts syntactically as arguments, and, at least potentially, both arguments and adjuncts as being syntactically optional[33]. Following overall LDSNL strategies, and similar to the Adjunction rule discussed above, the proposal involves the structural underspecification of lexical information. However, in contrast to the Adjunction rule, the underspecification is not located in the type information of the adjunct, but in the type information of the verb. The idea is that verbs structurally underspecify the number of Ty(e) expressions, taken to include both NPs and PPs, with which they may combine to form a verb

33 This view of adjunction has been explored in the Categorial Grammar paradigm notably by McConnell-Ginet (1982), whose analysis I discuss in the next chapter.

phrase. In addition to obligatorily required complements, the type information of verbs which I propose thus explicitly encodes the possibility of adding arguments optionally. If further Ty(e) expressions are introduced, they are treated as arguments. The argument adding is modelled as a step-by-step process, where the valency of the verb is increased in the context of suitable (in a sense left open at present) Ty(e) expressions, so that verb phrase interpretation can then be seen to proceed incrementally. One consequence of this view is that the arity of a predicate introduced by a verb is in general determined anew at every occasion of use of the verb.

In this section I provide the necessary formal definitions for this model of underspecification. I first define an underspecified type specification which I assume is lexically associated with verbs (Section 4.1.). The following sections illustrate how derivations with underspecified types for verbs can be formally defined. In Section 4.6., a sample derivation is provided as an illustration. The analysis is discussed and evaluated in the following Section 5.

4.1. Definition of Ty(e* –> X)

The lexical type specification associated with verbs in LDSNL is a conditional type specifying the number of Ty(e) expressions required by the verb to give an expression of Ty(t), as for example in Ty(e –> t) for intransitive verbs. I retain this general format, but add to it the possibility of introducing additional expressions of Ty(e). This is achieved by making use of the Kleene star operation, already employed in the system for the analysis of fronted constituents.

In particular, I define the Type value Ty(e* –> X) as in (19):

(19) *Definition of (e* –> X)*

$(e^* \rightarrow X) \quad =_{def.} \{(X) \vee (e^* \rightarrow (e \rightarrow X))\}$
where $X \in \{\text{TYPE}\}$

This reads as 'if estar, then some type X' is defined as 'X, or if estar, then if e, then X'. The net effect of this definition is that 'e*' is underspecified as to how many 'e's' it stands for. This is achieved by defining e* disjunctively, where the left part of the disjunction corresponds to the case where e* stands for zero e's, and the right part corresponds to the case where e* stands for one e. Since the definition is recursive (the right part of the disjunction is itself an e* predicate), e* can be interpreted as standing for an infinite number of e's[34]. Note that the definition in (19) does not enrich the vocabulary of the language Type, since e* is defined over elements of that language[35]. In fact, starred types can be defined for all types by the generalized star definition in (20):

(20) *Definition of (X* –> Y)*

$(X^* \to Y) \;=_{def.}\; \{(Y) \vee (X^* \to (X \to Y))\}$
where $X, Y \in \{\text{TYPE}\}$

I do not explore the possibilities the definition in (20) raises. In this thesis I only employ e* for verbs, so that the definition in (19) can be expressed more specifically as:

(21) *Definition of (e* –> t)*

$(e^* \to t) \;=_{def.}\; \{(t) \vee (e^* \to (e \to t))\}$

From the definition in (21) it follows that the underspecification inherent in the starred type can be resolved to predicates of any arity:

(22) *Resolution of Ty(e* –> t)*

$Ty(t)$
$Ty(e \to t)$
$Ty(e \to (e \to t))$
$Ty(e \to (e \to (e \to t)))$
$Ty(e \to (e \to (e \to (e \to t))))$
...

Note that after resolution of the underspecification, the starred type reduces to some ordinary type specification already employed in the LDSNL system, which interacts with binary branching trees equally employed in the system.

A possible alternative formulation would result in flat, n-ary branching trees. I introduce one such formalization here for illustration purposes, although I do not develop it further in what follows:

34 This is of course a very powerful characterization, a point which I discuss below.
35 The disjunction symbol (vel) in the definition is not part of the type, but meta vocabulary.

(23) *Definition of* $(e^n \to t)$

$(e^n \to t) =_{def.} \{(t) \vee ((e_1, e_2, \ldots e_n) \to t)\}$
where $n \in N$

The new aspect of (23) is that expressions of Ty(e) are added to an unordered set with little internal structure[36]. The disadvantage of this formalization is that it requires much more far reaching revisions of the LDSNL system than the alternative given in (21), since in (23) the language Type is enriched (by the comma), and corresponding tree relations are not defined. On the other hand, any advantages gained from either (21) or (23) can only be seen once the types are used, and the role of order of Ty(e) expressions for structure building and interpretation is explained. As it turns out, the formalization in (21) is completely adequate for the purposes I have in mind, so that I will in the remainder of this thesis only briefly point to differences between (21) and (23)[37].

4.2. Tree Building with e*: Introduction

In order to use the underspecified type e* in the LDSNL system, it has to be stated how transitions from one parse state to another are licensed when predicates with e* are involved. In particular, in order to integrate verbs with e* into a derivation, the DU with e* has to be associated with a correspondingly underspecified location in the tree. This means that transitions with e* can be characterized as subcases of transitions for underspecified tree nodes. There are two ways in which the association between the underspecified type e* and an underspecified tree location can be expressed with the resources given by the LDSNL architecture. Either a general transition rule is defined, which licenses derivations with e*, or derivations are driven by a set of actions, associated with predicates involving e*, which are stated in the lexicon. I discuss these alternatives in turn.

4.2.1. e* Introduction by Rule

The first alternative is to define a transition rule for e* similar to the transition rules outlined in Chapter 1, which licenses the introduction of a DU with e*

---

36 The internal structure in (23) is expressed by a comma. Equally well I could have used conjunctions, or required an ordered set.

37 In particular with respect to Minimal Recursion Semantics discussed in the next chapter.

type, and its association with an underspecified tree location. This can be done by employing the rules Introduction and Star Adjunction. An e^* Introduction rule is formulated in (24):

(24) *e^* Introduction*

$$\frac{\{_n \ldots ?\,Ty(X)\}}{\{_n \ldots ?\,Ty(X),\ ?\langle d^* \rangle\, Ty(e^* \rightarrow X)\}}$$

The rule states that at a parse state where at a given node Tn(n) TODO Ty(X) (where X any type) holds, a modal requirement may be introduced which requires that in a position lower than Tn(n) a DU with the underspecified type value $Ty(e^* \rightarrow X)$ (where X is the same type as at Tn(n)) be found. By Star Adjunction, an unfixed node may be built where the new DU is introduced. The corresponding tree structure resulting from e^* Introduction and Star Adjunction is given below:

(25) *e^* Introduction: Tree*

• $\{Tn(n), \ldots ?\,Ty(X)\}$

$\{Tn(n^*), \ldots, ?Ty(e^* \rightarrow X)\}$

The e^* Introduction rule ensures that verbs with an e^* type are introduced into the parse, by assigning them an underspecified tree node predicate $Tn(n^*)$. The type underspecification corresponds thus to an underspecification of location – it is as yet not known where in the eventual tree the DU from the verb is located. This situation means that there might be two locationally underspecified nodes in a given tree, since now both the verb and any unlocated nominal constituent (e.g. *Who*) are unfixed. However, the two nodes can always be distinguished by their type values, since constituents standardly analysed as being unfixed are Ty(e) expressions, while unfixed e^* expressions are always of $Ty(e^* \rightarrow t)$, so that no unwanted interaction between the two nodes results. This is because I restrict here underspecified types to verbs, so that the e^* Introduction rule can only be used in environments where the requirement is to build a Ty(t) expression from a given number of e's. Under this interpretation, the variable 'X' used in the rule does effectively stand for requirements such as $Ty(e \rightarrow t)$, $Ty(e \rightarrow (e \rightarrow t))$, etc.

However, there are two problems with defining a general rule for e^* introduction even if it is restricted to verbs. The first is that the rule is specifically designed for underspecified types which are meant to be associated with verbs. However, despite involving type underspecification, verbs might still vary as to their strictly required arguments. For example, *sing* might be of Ty(e* –> (e –> t)), but *close* of Ty(e* –> (e –> (e –> t))), but the rule is neutral with respect to this difference. Since idiosyncratic information of individual verbs has to be introduced from the lexicon, it seems to be more economical to also introduce their (underspecified) type value by lexical instruction, in which case a general introduction rule is unnecessary. In other words, the general aspect of verb phrase underspecification is that Ty(e) expressions may be introduced optionally, and this should be captured by a general rule. However, the contribution of the verb in this process is not in general optional, but rather to provide the (or, one) necessary condition for the process, which might also include idiosyncratic information. The general rule does not adequately express the contribution of the individual verb, which can be better stated as lexical instructions.

Secondly, the rule e* Introduction as it stands licenses the introduction of an underspecified location with an underspecified type at any parse stage. Although e* Introduction is, like all transition rules, optional, this treatment still seems to be unecessarily powerful. In a strict SVO language like English, for example, verbs are in general introduced at the VP node, i.e. at a node with the requirement TODO Ty(e –> t), which acts as a condition for the introduction of the verb from the lexcion, so that there are effectively no cases where the e* Introduction would be needed. On the other hand, a general rule for e* Adjunction might be needed in languages where the position of the verb is less predictable, as for example for an analysis of verb 'raising' in German. I discuss this point in more detail in Section 5, after the e* analysis has been more fully formulated, but I note here already that a general rule of e* Introduction threatens to undermine an analysis of unfixed verbs for the phenomenon characterized in other frameworks as verb movement, and that, since the introduction of unfixed nodes from the lexicon provides a viable alternative to introduction by rule, it seems more cautious to reserve the general rule option for alternative applications.

The conclusion is then that e* Introduction fails to address the contribution of individual verbs to the development of tree structure, while at the same time being too powerful for the aims it is designed to achieve. I thus consider in the next section an alternative formulation under which underspecified types are introduced into the tree from the lexical entry of verbs.

4.2.2. e* Introduction from the Lexicon

As outlined in Chapter 1, lexical entries in LDSNL consist of a set of statements which define the particular context in which a lexical expression may be introduced into a derivation, and how it contributes to the development of tree structure. For example, the lexical entry for *sing* can be stated, without e*, as follows:

(26) *Lexical Entry for* sing

IF ?Ty(e –> t)
THEN put(Fo(sing'), Ty(e –> t))
ELSE abort

According to (26), the condition for the introduction of the DU introduced by *sing* is a requirement TODO Ty(e –> t). If this condition is met, then the node with the requirement may be annotated with the formula and type values as stated in the entry. If the requirement is not met, the derivation ends. For the introduction of e*, the lexical information has to be modified, since the DU with the predicate has to be associated with an unfixed node. This is expressed in the lexical entry in (27):

(27) *Lexical Entry for* sing *with e**

IF ?Ty(e –> t)
THEN make(<d*>), put(Fo(sing'), Ty(e* –> (e –> t))),
go(<u*> ?Ty(e –> t))
ELSE abort

The difference between these two specifications is that in (26), the DU from *sing* is associated directly at the node with the requirement, while in (27), a new unfixed node is built, where the DU is associated. The go predicate indicates that the next lexical action proceeds from the requirement TODO Ty(e –> t), i.e. the pointer moves back to the point of origin[38]. The effect of this lexical specification is the same as that of the e* Introduction rule of the last section, so that the same tree results. However, the difference between the two characterizations shows with a transitive verb like *close*, the lexical entry for which is given below:

38 Throughout this thesis I assume that statements such as go(<u*> ?Ty(e –> t)) in the entry above are instructions for the pointer to move up in the tree to the *first* node with the relevant requirement. In Kempson et al. (1999) this is defined explicitly, but for the cases discussed here, the formulation adopted is sufficient.

(28) *Lexical Entry for* close *with e**

IF ?Ty(e -> t)
THEN make(<d*>), put(Fo(close'), Ty(e* -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t)),
put(?<d_1> Ty(e -> (e -> t))),
make(<d_0>), put(?Ty(e))
ELSE abort

In (28), it is part of the lexical specification of the verb to build the tree structure required needed to fulfill its subcategorization requirements. The entry licenses the building of three new nodes, an argument node (<d_0>), a functor node (<d_1>), both with the appropriate requirements, and an unfixed node for the verb. This means that both *sing* and *close,* and quite generally any verb, can be introduced as a requirement of TODO Ty(e -> t). The corresponding tree structures are shown below:

(29a) *Tree before the introduction of* close

• {Tn(0), ... ? Ty(t)}
/ \
• {Tn(00), ..., Ty(e)} • {Tn(01) ? Ty(e -> t) ◊}

The tree in (29a) includes the requirement TODO Ty(e -> t) at Tn(01), which provides a suitable condition for the introduction of *close.* The lexical specifications of *close* in (28) then result in the following tree:

(29b) *Tree after the introduction of* close

• {Tn(0), ... ?Ty(t)}
/ \
• {Tn(00), ..., Ty(e)} • {Tn(01) ?Ty(e -> t), ?<d_1> Ty(e -> (e -> t))}
/
• {Tn(010) ?Ty(e) ◊}

{Tn(01*), Fo(close'), Ty(e* -> (e -> (e -> t)))}

As can be seen from (29b), the lexical specifications specify all required nodes; the unfixed node Tn(01*) and the argument node Tn(010) are built directly, while the corresponding functor node is required by the modal statement holding at Tn(01), which results by the application of Prediction in the building of this node[39].

The same fomat for lexical information can be used for the ternary predicate *put*. Furthermore, since a lexical characterization can be used to encode specific idiosyncratic information provided by verbs, *put* may be characterized as requiring a Ty(e) expression referring to a location. A possible lexical specification for *put* is given below:

(30) *Lexical Entry for* put *with e* (first version)*

IF ?Ty(e -> t)
THEN make(<d*>), put(Fo(put'), Ty(e* -> (e -> (e -> (e -> t))))),
go(<u*> ?Ty(e -> t)),
make(<d_1>),
put(?Ty(e -> (e -> t)), ?<d_1> Ty(e -> (e -> (e -> t)))),
make(<d_0> +loc ?Ty(e)),
go(<u*> ?Ty(e -> t)),
make(<d_0>), put(? Ty(e))
ELSE abort

As in the previous entries, the entry for *put* in (30) specifies the tree structure to be built and introduces an unfixed node for the predicate, so that the condition for the introduction of *put* can be given as TODO Ty(e -> t). The specification of idiosyncratic requirements such as the '+loc' feature employed in (30) requires further discussion which is provided in the section on the role of prepositions below, where also a revised entry for *put* is proposed; the point here is rather that introducing e* in the lexicon has the additional advantage that highly specific information can be added by simple annotations.

The preceding discussion illustrates that the introduction of e* through lexical information, rather than by general rule, has the advantage that verbs in English can uniformly be introduced at a node requiring TODO Ty(e -> t), the VP node. Furthermore, in the lexical format precise statements regarding lexical idiosyncracies can be introduced, which are difficult to express by a gneral rule. In conclusion, then, I assume that underspecified type information involving e* is introduced via the lexical information from the given verb. In the next section I consider how the underspecification of e* predicates can be resolved.

39 In the following discussions I occasionally assume that Prediction applies automatically, i.e. I omit the step from requirement to node building.

4.3. Tree Building with e*: Resolution with Merge

Following the discussion in the last section, I assume that e* predicates are lexically associated with an underspecified tree node location. In this section, I show how the underspecified node is assigned its eventual fixed position in the tree, and how the underspecified type is resolved. The resolution of e*, however, is, in contrast to the introduction of e*, not subject to lexical requirements. As noted above, a rule characterization for the optionality of Ty(e) expressions is adequate. The problems with the introduction of e* by rule resulted from specific subcategorization information of individual verbs. Consequently, the resolution of e* can be defined by general rules. This section shows how e* interacts with the general rule Merge, which resolves the underspecification.

Since e* is defined as Kleene star operation, and since unfixed nodes are independently available in the system, the resolution of e* can be defined by exploiting independent LDSNL operations. In particular, an unfixed node with e* provides suitable input for the transition rule Merge, which is used in the resolution of unfixed nodes, for example with question words. As pointed out in Chapter 1, Merge is formulated completely generally, stating that two node descriptions (ND), corresponding to what is referred to as DU here, can be combined, so as to result in the union of whatever information is associated with the two DUs (Kempson, Meyer-Viol & Gabbay 1999: 82):

(31) *Merge*

$$\frac{\{\ldots \mathrm{ND}, \mathrm{ND}' \ldots\}}{\{\ldots \mathrm{ND} \cup \mathrm{ND}' \ldots\}}$$

The rule licenses effectively the merging of the information from two distinct nodes into one node. In practice, the effect of the rule is restricted since in most cases merging two nodes results in inconsistent information. However, in the case of underspecified nodes, it is exactly the combination of information which is required. Thus, in the case of e*, Merge is instantiated as follows:

(32) *Merge e**

$$\frac{\{_n \ldots ?\,Ty(X)\}, \{_{n^*} \ldots Ty(e^* \rightarrow X)\}}{\{_n \ldots Ty(e^* \rightarrow X)\ ?\,Ty(X)\}}$$

In this instantiation, the information from the unfixed node is merged with the requirement of a fixed node in the tree. If, by the definition of e* in (19), Ty(e* -> X) can be reduced to Ty(X) in this situation, Thinning can apply, and the requirement is fulfilled. The effect of Merge e* can be represented by two trees, corresponding to the input and to the output of Merge e* respectively:

(33a) *Merge e* (before)*

• {Tn(n), ... ? Ty(X)}

{Tn(n*), ..., Ty(e* -> X)}

Before the application of Merge, the e* predicate is at an unfixed node, and some requirement holds at a fixed node somewhere above the unfixed node. The application of Merge results in the merging of the information of the two nodes:

(33b) *Merge e* (after)*

• {Tn(n), ..., Ty(e* -> X), ?Ty(X)}

The resolution of Ty(e*) predicates is thus achieved in the same way that underspecified nodes are resolved in general, namely by Merge. Thus, for example, a derivation for *John sang* with e* involves the introduction of an unfixed node, by the lexical information from *sing*, and the resolution of the underspecification by Merge:

(34a) *Tree for* "John sang

```
                         • {Tn(0), ?Ty(t)}
                                  / \
• {Tn(00), Fo(john'), Ty(e)}        • {Tn(01), ?Ty(e -> t)}

{Tn(01*), Fo(sing'), Ty(e* -> (e -> t))}
```

The tree in (34a) is the result of building the tree structure specified in the lexical entry for *sing* in (27). In the absence of further input, Merge can apply to derive the tree in (34b):

(34b) *Tree for* "John sang"

```
                          • {Tn(0), ?Ty(t)}
                                  / \
• {Tn(00), Fo(john'), Ty(e)}   • {Tn(01), Fo(sing'),
                                 Ty(e* -> (e -> t)),
                                 ?Ty(e -> t)}
```

The information holding at Tn(01) is consistent, since Ty(e* -> (e -> t)) can be resolved as Ty(e -> t), and thus Thinning can apply, which then leads to the completion of the derivation.

4.4. Introduction of Optional Ty(e) Expressions

The initial and the final steps in derivations with e* have so far been defined: the introduction of e* predicates from the lexicon, and the resolution of the underspecification by Merge. What needs to be defined is the introduction of additional Ty(e) expressions, and how they interact with the predicate. Again, I discuss two options, the first employing the general rule Introduction, and the second, better one, employing lexical actions for prepositions.

4.4.1. Ty(e) Introduction by Introduction Rule

The optional intermediate steps in a derivation, in particular the optional introduction of Ty(e) expressions, could again be stated by referring to the rules Introduction and Prediction, which are independently defined in the system. Consider a tree with an unfixed e* node:

(35a) *Tree with unfixed e* node*

```
              • {Tn(n), ... ? Ty(X)}

{Tn(n*), ..., Ty(e* -> X)}
```

Scanning an incoming Ty(e) expression, the tree is updated by Introduction of the requirement TODO (<d> ?Ty(e)) and the building of the requisite node by Prediction:

(35b)

```
                         • {Tn(n), ... ? Ty(X)}
                              /
      • {Tn(n0), ..., Ty(e)}

    {Tn(n*), ..., Ty(e* -> X)}
```

Similarly, Introduction and Prediction result in the building of a new functor node with the requirement Ty(e -> X):

(35c)

```
                         • {Tn(n), ... ? Ty(X)}
                              / \
      • {Tn(n0), ..., Ty(e)}   • {Tn(n1) ? Ty(e -> X)}

    {Tn(n*), ..., Ty(e* -> X)}
```

Two further applications of these two rules would result in two new nodes, one of which carries a requirement of Ty(e -> (e -> X)). This process could go on until no further Ty(e) expressions are scanned. At every step, Merge could apply to resolve the underspecification. This solution would drive derivations with e* with no changes to the overall system – the introduction of the DU with e* at an unfixed node would be followed by possibly several applications of Introduction and Prediction, and the resolution of the underspecification by Merge.

However, this solution threatens to be too powerful. Lexical information merely requires the presence of a given number of Ty(e) expressions. If further Ty(e) expressions could be freely introduced without any restrictions, as is indeed the case in this solution with Introduction, there would be no means to exclude strings like (36):

(36a) ?John met Mary Bill the house.

(36b) ?Sally closed the window the kitchen.

(36c) ?Allan put the flowers the table.

The strings in (36) would go through since the noun phrases are introduced by Introduction, and e* allows for the introduction of Ty(e) expressions into the predicate. However, what is missing in (36) is a preposition:

(37a) John met Mary with Bill in the house.
(37b) Sally closed the window in the kitchen.
(37c) Allan put the flowers under the table.

From the contrast between (36) and (37), it appears that prepositions function to license the introduction of further Ty(e) expressions. In the next section, I discuss how this function can be best expressed in the present context, and how the analysis of prepositions can be used to provide an alternative to the analysis employing the Introduction rule.

4.4.2. Ty(e) Introduction from the Lexicon

I have so far at least tacitly assumed that PPs are of Ty(e). It is only under this assumption that an e* analysis could be employed for PPs which behave as an argument of the verb:

(38a) Fran opens bottles with her teeth.
(38b) The Bakers travelled to Paris in their van.
(38c) They discovered the corpse in the fridge.

The idea is that the verb is underspecified, and that all post-verbal constituents in (38) can be introduced as arguments to the predicate[40]. Given the definition of e*, both NPs and PPs have to be of Ty(e) for this analysis to work. But this state of affairs, in conjunction with the unrestricted availability of Introduction, leads to the undesirable results illustrated at the end of the last section, namely that not only PPs, but both PPs and NPs can freely be added into the VP, which by and large is not true.

One possible solution to this problem would be to say that PPs are not of Ty(e) after all, but of a discrete type, say, for the sake of argument, of Ty(pp), so that it is not Ty(e) expressions but Ty(pp) expressions which can be freely added and be incorporated into the predicate by a corresponding type, say, Ty(pp* –> (e –> t)). But that would take almost all of the original motivation for e* out of the picture since it is exactly PPs which appear most clearly to modify verbs in an argument-like fashion and thus should be treated in a similar way. In addition, it would increase the available types in LDSNL without, I believe, proper motivation.

40 In the relevant reading, that is. PPs may modify NPs, a case discussed below.

A better way to go seems to be to make prepositions the licensors of Introduction, for example by saying that Ty(e) expressions can only be introduced if they are marked by a preposition. But that runs into problems for the main context where Introduction is needed. This is the building of the subject node in English. Introduction licenses the transition from a requirement TODO Ty(t) to the building of a Ty(e) node:

(39a) • {Tn(0) ? Ty(t)}

The application of Introduction and Prediction at a parse stage like (39a) results in a tree such as (39b):

(39b) • {Tn(0) ? Ty(t)}

/

• {Tn(00) ? Ty(e)}

The rule Introduction is needed to ensure that an initial Ty(e) expression in a string of words can be assigned a position in the tree. Clearly, in this environment, the Ty(e) expression, i.e. the subject, need not, in fact cannot, be licensed by a preposition. Since I want Introduction to take care of the structure building in cases like (39), I cannot restrict it to be licensed by prepositions. This means that I cannot use Introduction for the introduction of optional Ty(e) expressions, and that optional Ty(e) expressions have to be introduced into the tree by some other means. I thus propose that optional Ty(e) expressions are introduced by the lexical instructions of prepositions, and not by a general rule.

Consider a case like (40):

(40) Sandy opened the window with the broom.

Given the lexical defininitions for verbs above, the NP *the window* is licensed in the tree since the Ty(e) node has been built from lexical instructions. Furthermore, the predicate has been assigned to an unfixed node, resulting in a parse state like (41):

(41a) *Tree for* "Sandy opened the window

```
                          • {Tn(0), ?Ty(t)}
                                 / \
 • {Tn(00), Fo(sandy'), Ty(e)}  • {Tn(01), ?Ty(e -> t)}
                                         / \
    • {Tn(010), Fo(the_window'),     • {Tn(011),
                         Ty(e)}        ?Ty(e -> (e -> t)}

{Tn(01*), Fo(open'), Ty(e* -> (e -> (e -> t)))}
```

The next step in the derivation should result in the introduction of the PP at a new argument node Tn(0110). I assume for the moment that Introduction is blocked from applying in this situation (in a sense to be made more precise soon). The sensible assumption then is that it is the preposition *with* which builts the necessary node, to result in a tree like (41b):

(41b) *Tree for* "Sandy opened the window with

```
                          • {Tn(0), ?Ty(t)}
                                 / \
 • {Tn(00), Fo(sandy'), Ty(e)}  • {Tn(01), ?Ty(e -> t)}
                                         / \
    • {Tn(010), Fo(the_window'),     • {Tn(011),
                         Ty(e)}        ?Ty(e -> (e -> t)}
                                              /
                          • {Tn(0110), ?Ty(e)}

{Tn(01*), Fo(open'), Ty(e* -> (e -> (e -> t)))}
```

The tree in (41b) is identical to a potential tree where Tn(0110) would have been introduced by Introduction. Thus, it can be developed further without problems. Prediction results in the building of a functor node Tn(0111) with Ty(e -> (e -> (e -> t))), and the DU from *the broom*, an expression of Ty(e), fulfills the requirement at Tn(0110). Finally, Merge applies, and the derivation is finished.

The claim is thus that Tn(0110) is built not by Introduction but from the lexical information from the preposition *with*. This can be achieved with the following lexical entry:

(42) *Lexical Entry for* with *(first version)*

IF ?Ty(e –> (e –> t))
THEN put(?$<d_1>$ Ty(e –> (e –> (e –> t)))),
make($<d_0>$), put(?Ty(e))
ELSE abort

While some modifications are still needed, it is clear that this specification does what is needed – the entry specifies that *with* builds an argument node with the requirement TODO Ty(e). I assume that lexical entries for prepositions consists minimally of such an instruction to build a TODO Ty(e) node. There are two further points. The first concerns the condition (i.e. IF) for the introduction of the preposition. In (42), the preposition can be introduced if there is a requirement TODO Ty(e –> (e –> t)) at the relevant parse state, as was the case in the example discussed (i.e. (40)). In general, however, PPs can be introduced at any stage with the requirement for a predicate, e.g. TODO Ty(e –> t), TODO Ty(e –> (e –> t)), TODO Ty(e –> (e –> (e –> t))) etc. But this is exactly the range of cases covered by Ty(e* –> t). Thus e* can be used here as a requirement. Given that the relevant PPs here are introduced into the VP, the universal condition for the introduction of prepositions is Ty(e* –> (e –> t)). This results in the following first version of a schematic entry for prepositions, which will be modified below:

(43) *Schematic Lexical Entry for Prepositions (first version)*

IF ?Ty(e* –> (e –> t))
THEN put(?$<d_1>$ Ty(e –> (e –> t))),
make($<d_0>$), put(?Ty(e))
ELSE abort

With this specification, the Ty(e) node needed to drive intermediate steps in derivations with e* is built by lexical instructions from the lexical entries of prepositions, without the need for the Introduction rule. Note that the requirement obtaining at the argument daughter here implies the resolution of the underspecified requirement; in particular, it has to be read not as an absolute statement, but as the resolved type specification relative to the parse state. Thus for a requirement such as Ty(e* –> (e –> (e –> t))), the first action is put(?$<d_1>$ Ty(e –> (e –> (e –> t)))). In this way, the preposition requires the building of a corresponding functor node by Prediction.

A second point relevant for the lexical entry of prepositions is their 'semantic' contribution, since the entry thus far only specifies that prepositions build nodes. I will here not provide a detailed discussion of the meaning of prepositions in English (or in general). Rather I simply show how the semantic contribution of prepositions can be encoded into lexical entries of the kind discussed here in general. The easiest way to do this is to employ features which specify that the Ty(e) expressions be of a particular semantic/thematic kind. In the example considered above, for example, *with the broom* may be regarded as an 'instrument', so that the preposition in this case requires the Ty(e) node to be annotated with a feature '+instr'. This may simply be written into the lexical specification:

(44) *Lexical Entry for* with *(final version)*

IF	?Ty(e* -> (e -> t))
THEN	put(?<d_1> Ty(e -> (e -> t))),
	make(<d_0>), put(+instr, ?Ty(e))
ELSE	abort

A feature like '+instr' in (44) can be thought of as a language specific set of words, e.g. the feature '+loc' in English would be something like {in, on top of, behind, under, ... }, or it can be given a specific semantic interpretation, e.g. in terms of force dynamics, or lexical-conceptual structures. However, I am not developing such an account here, nor do I review the available literature. Although formal semantic accounts have been developed for temporal and spatial prepositions[41], the analysis of instruments or commitatives as in (45) appears to be much more intricate:

(45a) Fran wrote her dissertation with an ink pen.

(45b) John went to the party with Mary.

Rather than defining the particular semantic contribution of the preposition in cases like these, I simply use features, and, since I do not go into further details here, I prefer to use one feature per preposition, that is, for example, *with* annotates the Ty(e) node with a 'feature' '+with'. On occasion, I use a general feature '+prep', meaning a Ty(e) expression with a preposition, i.e. a PP. For the purposes here, the main contribution of prepositions in cases where PPs function as arguments to the verb is to license the introduction of their object (i.e. the noun phrase) into the predicational range of the verb, that is, prepositions may function, in conjunction with the verb, as instructions for

41 For example, Dowty (1979), Gawron (1985), Pratt & Francis (1997), Zwarts (1997), as well as several non-formal approaches, e.g. Bennett (1975), Jackendoff (1983).

argument adding. In this sense, the use of prepositions corresponds to the function of case in languages like Finnish or German, which mainly signals the relation of the noun phrase with the verb without any clear semantic contribution, at least with respect to tree building, which is the focus of this chapter[42]. In the cases in (45), for example, the contribution of the prepositions is intuitively to allow the construction of a three place semantic writing relation between Fran, her dissertation, and an ink pen for (45a), and the construction of a three place semantic going relation between John, the party, and Mary for (45b). This process of constructing semantic relations will be discussed more extensively in the next two chapters. For the present, I assume that the information from the preposition is projected into the formula value of the Ty(e) expression, so that *with the broom* results in a Ty(e) expression with a formula value Fo(with_the_broom). The interpretation of formula values in general will be discussed in more detail in the following two chapters, so that I assume here that prepositions minimally build a Ty(e) node and provide some annotation on it.

The lexical treatment of the introduction of a Ty(e) node provides an alternative to the introduction of Ty(e) expressions by Introduction discussed in the last section. However, the analysis is not yet a solution to the problem of the unwanted optional NPs. This is because the application of general rules is in general optional. Thus, although Ty(e) nodes can now be built from the lexicon, they still can be built by Introduction. What is missing is a statement to the effect that Ty(e) expressions can *only* be built from the lexicon, in other words, the assumption that Introduction is blocked which I have made in this section has to hold. It is, however, difficult to block a rule from applying since rules, including Introduction, are not context sensitive in the relevant sense. Rather, what this analysis of e* entails is that there is no general rule Introduction. As pointed out above, Introduction is needed for the introduction of subjects in English. But this is, as far as I can see, the only context where the rule is required. If this turns out to be true, Introduction can be replaced by (46):

(46) *Subject Introduction*

$$\frac{\{_n \ldots ?Ty(t)\ \Diamond\}}{\{_n \ldots ?Ty(t),\ ?\langle d_0\rangle\, Ty(e),\ ?\langle d_1\rangle\, Ty(e \rightarrow t)\ \Diamond\ \}}$$

---

42 This position seems also plausible with respect to language change, since it has been noted especially in the Grammaticalization literature that case markers diachronically develop out of prepositions; cf. e.g. Heine, Claudi & Hünnemeyer (1991), Hopper & Traugott (1993).

The rule Subject Introduction licenses the introduction of a Ty(e) node, which then can be built by Prediction, only in context of a Ty(t) task, that is, at the root node. This still licenses the introduction of subjects (in languages where this is necessary), but does not interfere with the introduction of Ty(e) expressions at any other position in the tree. This is in fact the only major revision of the overall LDSNL architecture which the analysis of adjunction by e* developed here requires, and for the purposes of this thesis I assume that Introduction is replaced by Subject Introduction.

Before progressing, I briefly mention two implications of this analysis of prepositions. The first concerns the lexical entry for verbs like *put*, obligatorily requiring a PP. The tentative lexical entry for *put* developed in Section 4.2.2. is repeated here for convenience:

(47) *Lexical Entry for* put *with e* (first version)*

IF ?Ty(e -> t)
THEN make(<d*>), put(Fo(put'), Ty(e* -> (e -> (e -> (e -> t))))),
go(<u*> ?Ty(e -> t)),
make(<d_1>),
put(?Ty(e -> (e -> t)), ?<d_1> Ty(e -> (e -> (e -> t)))),
make(<d_0> +loc, ?Ty(e)),
go(<u*> ?Ty(e -> t)),
make(<d_0>), put(? Ty(e))
ELSE abort

The relevant clause in (47) is make(<d_0> +loc, ?Ty(e)). In the light of the discussion in this section, this has to be revised, since all properties of the relevant node can now be built from the preposition. However, the requirement for such a node to be in the tree is still part of the lexical information from *put*. The revised entry is as follows:

(48) *Lexical Entry for* put *with e* (final version)*

IF ?Ty(e -> t)
THEN make(<d*>), put(Fo(put'), Ty(e* -> (e -> (e -> (e -> t))))),
go(<u*> ?Ty(e -> t)),
make(<d_1>), put(?Ty(e -> (e -> t)), ?<d_0> +prep, Ty(e),
?<d_1> Ty(e -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t))
make(<d_0>), put(? Ty(e))
ELSE abort

The revised entry states that there be a Ty(e) node with an annotation +prep as a requirement at the VP node. That is, no node is built, but its presence is required[43]. Again, there is some uncertainty about the feature value. The way the entry reads now results in a successful derivation of, for example, (49):

(49) ?Sally put the books at noon.

The question is whether (49) is ungrammatical or just semantically unacceptable. According to (48), (49) is grammatical, because I have opted for a more liberal characterization of the lexical syntactic requirements. Alternatively, one might use a feature +loc again, as in (47), which would block (49). Since I am here concerned with general characterizations of tree growth, I leave the question open. The important point here is that in a derivation invloving *put* and a PP, some tree building comes from the lexical entry from *put*, and some from the preposition, where the latter is restricted by the lexical requirements imposed by *put*. Subcategorization can thus be characterized as either building nodes, or merely requiring nodes to be present.

A second implication of this analysis is that some NPs may lexically be characterized as building nodes. I am thinking here of adverbials like *yesterday*, *tomorrow*, or possibly *home*. For these, their lexical entries may specify that they effectively are like PPs without preposition, so that they may be freely introduced:

(50a) I saw him yesterday.
(50b) They will leave from Manchester tomorrow.
(50c) I walked her home.

A lexical characterization in the Ty(e) expressions in these examples guarantees that they, at least under some readings, can be introduced as optional arguments to a given predicate. However, rather than going into more details here, I leave this as a possibility, and continue with the characterization of derivations with e*.

In this section, I have argued that optional Ty(e) expressions are introduced into the tree structure by instructions from the lexicon. This treatment has the advantage that the intuition that prepositions function as licensors of Ty(e) expressions is formally expressed. This analysis replaces a

43 The node can still be built by an (optional) application of Prediction, but this would result in an illicit derivation since there would be two nodes requiring TODO Ty(e).

putative alternative under which Ty(e) nodes can freely be built by the general rule Introduction. A derivation with e* thus involves the building of an unfixed node for the predicate, which is achieved by lexical instructions, the optional building of Ty(e) nodes by prepositions which are filled by their NP objects, and finally, the application of Merge which results in the resolution of the underspecification of the type, and the assignment of a fixed position in the tree of the unfixed node. Note that under this characterization, the underspecified type is resolved 'in one go', i.e. only with the application of Merge is the underspecified type resolved to whatever the current requirement is. Although I think this is ultimately correct, I develop in the next section a rule which allows for the incremental partial resolution of e* predicates, where every introduction of an expression of Ty(e) is 'registered' at the unfixed node. This incremental notion of e* resolution is needed for the semantic analysis developed in the next chapter, although it is not needed for the conceptual analysis developed in Chapter 5.

4.5. Incremental Transition Rule for e*

As just noted, the definitions discussed in the preceding sections imply that the underspecification of e* is resolved at once, with the application of Merge. If, for example, there is a node with the requirement Ty(e –> (e –> (e –> t))), the underspecified type Ty(e* –> t) can be resolved as meeting this requirement. However, for some applications, it might be useful to have a more context sensitive rule for e* which states that the underspecification is partially resolved with every instance of the introduction of a Ty(e) expression. The choice depends really on the interpretation of underspecifed verbs, which is discussed in the next two chapters. In this chapter, I formulate a rule which allows for partial resolution of e* predicates as a more specific characterization of its syntax.

The rule for partial resolution applies after the introduction of e* and before the application of Merge and can be stated as follows[44]:

(51) *e* Partial Resolution*

$$\frac{\{_{n^*} \ldots Ty(e^* \text{->} X)\}, \{_{n0} \ldots Ty(e) \Diamond\}}{\{_{n1^*} \ldots Ty(e^* \text{->} (e \text{->} X)\}, \{_{n0} \ldots Ty(e) \Diamond\}}$$

44 The rule could be regarded as an instantiation of the rule Propagation, which ensures that an unfixed node is 'passed down' the tree. However, Propagation does not define the updating of the type value, which is expressed in the rule given here.

The rule licenses the following transition: Given a parse state with a DU with underspecified type Ty(e* -> X) at an underspecified location Tn(n*) and the current task state at an argument node where a DU of Ty(e) holds, the underspecified node can be updated to 1) holding at or below the corresponding functor node Tn(n1), and 2) being minimally reducible to an expression with Ty(e -> X), but not simply to Ty(X). According this rule, the introduction of the Ty(e) expression is 'registered' at the unfixed node by partially resolving the underspecification. The following trees illustrates the transition licensed by e* Partial Resolution:

(52a) *e* Partial Resolution: Tree (before)*

```
                              • {Tn(n), ... ?Ty(X)}
                             /
        {Tn(n0), ..., Ty(e) ◊}•

{Tn(n*), ..., Ty(e* -> X)}
```

(52b) *e* Partial Resolution: Tree (after)*

```
                              • {Tn(n), ... ?Ty(X)}
                             /
        {Tn(n0), ..., Ty(e) ◊}•

{Tn(n1*), ..., Ty(e* -> (e -> X)}
```

The effect of the rule is to update the information at the unfixed node. The rule may apply after every introduction of a Ty(e) expression into the tree. The final resolution of the underspecification of e* is achieved by Merge, as before.

In what follows, I assume that e* Partial Resolution is part of the LDSNL rules until the interpretation of e* predicates has been discussed. Since this is a more specific statement of the syntax of e*, any derivation with the rule is also licensed without it, so that no harm is done to assume it.

With these formal specifications at hand, the function of e* within the LDSNL model can be demonstrated by a sample derivation provided in the next section.

4.6. e* for Verbs: Sample Derivation

The combination of the definition of e*, the lexical specifications, and the rules discussed in the preceding sections together define complete derivations with e* predicates. In this section I provide a sample derivation, illustrating how e* is used in the LDSNL system. I assume here that at least one Ty(e) expression is lexically required by every verb, a point discussed in more detail in the next section. On the other hand, I assume that all verbs allow for the introduction of optional Ty(e) expressions. Thus, the lexical entry for the (transitive)[45] verb *bake* can be represented as in (53):

(53) *Lexical Entry for* bake *with e**

```
IF      ?Ty(e -> t)
THEN    make(<d*>), put(Fo(bake'), Ty(e* -> (e -> (e -> t)))),
        go(<u*> ?Ty(e -> t)),
        put(? <d1> Ty(e -> (e -> t))),
        make(<d0>), put(?Ty(e))
ELSE    abort
```

The important part of the entry in this context is the underspecified type value Ty(e* -> (e -> t)), which states that the DU introduced by *bake* minimally requires one expression of Ty(e) (the 'subject'), and that it allows for a potentially unlimited number of optional Ty(e) arguments.

In this section, I give a sample derivation of the following example in (54), involving *bake,* to show how the definitions for e* are used in the unfolding tree structure:

(54) John was baking a cake for Mary in the kitchen.

The example in (54) involves two prepositional VP adjuncts, the 'benefactive' *for Mary* and the locative *in the kitchen.* The relevant derivational steps are as follows:

(55a) *Tree for* "John

```
                 • {Tn(0), ?Ty(t)}
                        / \
• {Tn(00), Fo(john'),      • {Tn(01), ?Ty(e -> t)}
   Ty(e)}
```

The first steps in the derivation up to the scanning of the verb are familiar. The requirement TODO Ty(e -> t) holding at Tn(01) is the condition for introducing the information from *bake.* Thus, two fixed daughter nodes are built, as well as an unfixed node for Fo(bake'):

45 Cf. the section on optional arguments below.

(55b) *Tree for* "John was baking

```
                         • {Tn(0), ?Ty(t)}
                                  / \
• {Tn(00), Fo(john'), Ty(e)}     • {Tn(01), ?Ty(e –> t)}
                                        /\
              • {Tn(010), ?Ty(e)}    • {Tn(011), ?Ty(e –> (e –> t))}

{Tn(01*), Fo(bake'), Ty(e* –> (e –> (e –> t))}
```

The next expression in the string, *a cake,* (I ignore the internal structure of the NP throughout) is of Ty(e) and thus fulfills the requirement holding at Tn(010):

(55c) *Tree for* "John was baking a cake

```
                         • {Tn(0), ?Ty(t)}
                                  / \
• {Tn(00), Fo(john'), Ty(e)}     • {Tn(01), ?Ty(e –> t)}
                                        /\
   • {Tn(010), Fo(a_cake), Ty(e)}    • {Tn(011), ?Ty(e –> (e –> t))}

{Tn(01*), Fo(bake'), Ty(e* –> (e –> (e –> t))}
```

At this stage in the derivation, Merge could apply, so that the underspecification would be resolved and the unfixed node be merged with Tn(011). This would in fact be the derivation of *John was baking a cake.* However, in this case, there is further input and the application of Merge is prevented due to the the lexical information from the next word, *for,* the lexical entry for which is:

(56) *Lexical Entry for* for

IF	?Ty(e* –> (e –> t))
THEN	put(?<d_1> Ty(e –> (e –> t))), make(<d_0>), put(+for, ?Ty(e))
ELSE	abort

If Merge applied before the introduction of *for,* the condition for introducing *for* would not be met, since there would be no requirement TODO Ty(e* –> (e –> t)), so that, by the ELSE clause, the derivation would be aborted. Since the application of Merge is optional, it is enough to say that here, it doesn't apply without any need to specify this further[46]. The information from *for*, then, results in the building of a new argument node:

(55d) *Tree for* "John was baking a cake for

• {Tn(0), ?Ty(t)}
/ \
• {Tn(00), Fo(john'), Ty(e)} • {Tn(01), ?Ty(e -> t)}
/\
• {Tn(010), Fo(a_cake), Ty(e)} • {Tn(011), ?Ty(e -> (e -> t))}
/
• {Tn(0110), +for, ?Ty(e)}

{Tn(01*), Fo(bake'), Ty(e* -> (e -> (e -> t))}

The new node Tn(0110) is built and annotated according to the lexical instructions from the preposition. The corresponding functor node Tn(0111) can be built by Prediction, and the next word in the string, *Mary*, of Ty(e), can be associated at Tn(0110), resulting in the following state:

(55e) *Tree for* "John was baking a cake for Mary

• {Tn(0) ? Ty(t)}
/ \
• {Tn(00), Fo(john'), Ty(e)} • {Tn(01) ? Ty(e -> t)}
/\
• {Tn(010), Fo(a_cake), Ty(e)} • {Tn(011) ? Ty(e -> (e -> t))}
/\
• {Tn(0110), Ty(e), Fo(for_Mary')} • {Tn(0111) ? Ty(e -> (e -> (e -> t)))}

{Tn(01*), Fo(bake'), Ty(e* -> (e -> (e -> t))}

At this stage, e* Partial Resolution can apply to partially update the unfixed node[47], which then becomes (I reproduce only the unfixed node here, the tree does not change):

(55f) {Tn(0111*), Fo(bake'), Ty(e* -> (e -> (e -> (e -> t)))}

46 Note that this works here, since the application of Merge would result in the application of the ELSE clause in the entry of *with*. The problem with the rule Introduction discussed above is different since its unrestricted use would not automatically result in abort.

47 Recall that I assume the rule e* Partial Resolution to be operative. Without it, nothing would happen at the unfixed node at this juncture.

By e* Partial Resolution, the tree node location is updated, and one e is introduced into the underspecified type, corresponding to the Ty(e) expression at Tn(0110) in the tree. The introduction of the second PP *in the kitchen* repeats these steps exactly – the preposition builds a node which is filled by the Ty(e) expression *the kitchen*, Prediction builds a functor node, and the unfixed node is updated by e* Partial Resolution. Of course, if there were no further input after *Mary*, the parse of *John was baking a cake for Mary* would be successfully completed by the application of Merge. With the introduction of *in the kitchen*, however, the resulting tree is:

(55g) *Tree for* "John was baking a cake for Mary in the kitchen

```
                    • {Tn(0), ?Ty(t)}
                            / \
• {Tn(00), Fo(john'), Ty(e)}   • {Tn(01), ?Ty(e -> t)}
                                       / \
  • {Tn(010), Fo(a_cake'), Ty(e)}   • {Tn(011), ?Ty(e -> (e -> t))}
                                             / \
    • {Tn(0110), Fo(for_mary'), Ty(e)}   • {Tn(0111),
                                          ?Ty(e -> (e -> (e -> t)))}
                                                / \
• {Tn(01110), Fo(in_the_kitchen'), Ty(e)}   • {Tn(01111),
                                          ?Ty(e -> (e -> (e -> (e -> t))))}

{Tn(n01111*), Fo(bake'), Ty(e* -> (e -> (e -> (e -> (e -> t)))))}
```

The tree in (55g) represents the parse state after introducing *in the kitchen*, the building of Tn(01111), and the updating of the underspecified type value. Since there is no further input, Merge applies and the underspecification is resolved in the manner defined in the definition of e*. The DU provided by *bake* is then associated at Tn(01111), fulfilling the requirement holding at that node, and Completion can apply:

(55h) *Tree for* "John was baking a cake for Mary in the kitchen"

```
                    •{Tn(0),
Fo((((bake'(in_the_kitchen'))(for_mary'))(a_cake'))(john')), Ty(t)}
                              / \
• {Tn(00), Fo(john'), Ty(e)}   • {Tn(01),
                                 Ty(e -> t)
          Fo(((bake'(in_the_kitchen'))(for_mary'))(a_cake'))}
                                  / \
  • {Tn(010), Fo(a_cake'), Ty(e)}   • {Tn(011),
                                      Ty(e -> (e -> t)),
                    Fo((bake'(in_the_kitchen'))(for_mary'))}
                                        / \
    • {Tn(0110), Fo(for_mary'), Ty(e)}    • {Tn(0111),
                                            Ty(e -> (e -> (e -> t))),
                                          Fo(bake'(in_the_kitchen'))}
                                        / \
• {Tn(01110), Fo(in_the_kitchen'), Ty(e)}   • {Tn(01111),
                                            Fo(bake'),
                                         Ty(e -> (e -> (e -> (e -> t))))}
```

The tree in (55h) concludes the derivation of *John was baking a cake for Mary in the kitchen* with the underspecified type value e*. I have assumed that the lexical type information of *bake* is underspecified and that it is resolved in the context of the utterance within which it is used. The definition of e*, together with lexical information and transition rules, results in a parse where Ty(e) expressions can be introduced into the tree as arguments to the underspecified verb, and where the underspecification is incrementally resolved, depending on the input to the parse.

However, I have not yet discussed any semantic interpretation for e*, which will be the topic of the following two chapters. There are, on the other hand, a number of points which need to be addressed before the discussion of semantic interpretation, so that I turn now to the discussion of some details of this analysis.

5. Discussion

In this section I discuss a number of questions related to the e* analysis developed in the preceding section and turn to points which have already been raised but have not been discussed in full. The first two sections are concerned with subjects and optional arguments, while the following section discusses the relation between the analysis proposed here and the LINK analysis of secondary predication proposed in Swinburne (1999). Section 5.4. offers several examples of how the analysis represents extraction, while the final Section 5.5. briefly points out how the analysis relates to positionally unfixed verbs such as are found in German.

5.1. Universal Subjects

As indicated above, I assume that verbs universally require at least one expression of Ty(e), corresponding to the logical subject. This does not follow, however, from the definition of e* given in (21). In fact, it is tempting to think of an example like *sing* that it lexically encodes the weaker type Ty(e* –> t), so that the following parse could be derived:

(56) • {Tn(0), Fo(sing'), Ty(t)}

The 'tree' in (56) could be thought of, for example, as the representation of imperatives like *Sing!*, since no element of type Ty(e) is in the utterance, and the underspecification of *sing* could, under this assumption, simply be resolved to Ty(t). The problem in the structure in (56) is that there is no expression in the tree corresponding to the logical subject – in the case of imperatives, the addressee. The preferred parse tree for *Sing!* is rather:

(57) • {Tn(0), Fo(sing'($U_{addressee}$)), Ty(t)}

/\

• {Tn(00), Fo($U_{addressee}$), Ty(e)} • {Tn(01), Fo(sing'), Ty(e –> t)}

In (57), the subject position is annotated with a formula value which has a metavariable, indicating an instruction to the hearer to search for a suitable representation, and a restriction on that variable to the effect that the interpretation of the metavariable be the representation of the addressee. For this example, imperatives in English, the licensing and annotating of the subject node involves several sources of information; the formula value results presumably from the information from the paradigm[48], while the licensing of the node as such might be regarded as resulting from an Introduction rule such as Subject Introduction. Yet intuitively, the presence of the subject node results partly from verbal information. Furthermore, on the technical side, it is not clear if the introduction of one 'e' into the

48 For example, 'mood', or 'clausal typing'; in fact this might be even more clearly expressed by a category feature similar to the Cat(+Q) feature in questions, e.g. as Cat(+Imp).

underspecified type (i.e. the application of the right disjunct of the definition) is licensed by the node built due to Introduction, or by the grammatical information from the paradigm, or both. To avoid this problem, and to capture the intuition about the relation between information from the verb and the subject, I restrict the underspecification in verbal subcategorization information to the verb phrase, that is, including objects and VP adjuncts, but excluding subjects. This is partly in order to restrict the scope of inquiry in this thesis, but also because I think it is correct: verbal information includes the requirement of the presence of a subject. I thus assume that all lexical specifications for verbs include a statement that at least one Ty(e) expression, directly dominated by the node with Ty(t), which is the position of the logical subject (cf. Kempson, Meyer-Viol, Gabbay 1999), be present[49].

5.2. Optional Arguments

Under the analysis developed here, Ty(e) expressions can be introduced into the verb phrase either by lexical instructions from the verb, or by lexical instructions from prepositions, and possibly, in some restricted cases, by lexical specification of bare Ty(e) adverbs. However, this means that optional arguments such as in (58 - 60) are not covered by the analysis:

(58a) Kelly was singing.

(58b) Kelly was singing a song.

(59a) Mr. Yu spent the whole afternoon baking.

(59b) Mr. Yu spent the whole afternoon baking cakes.

(60) Everybody was eating, but only Billy was eating pasta.

The optional objects in these examples are not introduced by a preposition, and they are not of the kind such that one would want to say that they build their own node by lexical specification. This leaves, under the proposal developed here, only the possibility that optional arguments are in fact introduced by lexical specification from the verb. This is, I think, partly warranted by the fact

49 Two further related problems are not discussed in this thesis: the analysis of 'weather' verbs which possibly do not have a logical subject; and the much bigger problem of grammatical functions, including grammatical function changing, i.e. the relation between grammatical and logical subjects, and grammatical expressions which appear to encode/change this relation. Both problems have not been addressed in detail, to my knowledge, in the LDSNL literature in general.

that in general there are semantic restrictions on which optional arguments go with which predicate, whereas these restrictions are much weaker for true optional Ty(e) expressions. I will thus assume here, that optional arguments are encoded in the lexical specifications of verbs. This means that a verb like *sing* has a disjunctive lexical entry, specifying the intransitive and the transitive use:

(61) *Disjunctive Lexical Entry for* sing *with e**

IF	?Ty(e -> t)
THEN	make(<d*>), put(Fo(sing'), Ty(e* -> (e -> t))),
	go(<u*> ?Ty(e -> t))
OR	make(<d*>), put(Fo(sing'), Ty(e* -> (e -> (e -> t)))),
	go(<u*> ?Ty(e)),
	put(? <d_1> ?Ty(e -> (e -> t))),
	make(<d_0>), put(?Ty(e))
ELSE	abort

As was the case with prepositions, the argument node may be annotated with further restrictions specifying that the expression of Ty(e) denote something singable, e.g. a song, or an aria. However, for the present, I leave the specification as it is as an illustration of a disjunctive entry.

This analysis of optional arguments as disjunctive lexical entries is in fact not quite as desired, since optional arguments can be regarded as a clear case of verbal underspecification, and they should thus be more amenable to the general analysis proposed here. However, I leave this question for future research, and keep the lexical analysis.

5.3. e* and LINK

Under the e* analysis, prepositions contribute to the building of tree structure, and provide an annotation of an associated Ty(e) expression. Prepositional phrases thus relate to verbs as arguments, rather than as functors. In this section, I discuss the relation between PPs as arguments and as functors in the light of the LDSNL operations LINK and e*.

The view that PPs are arguments of Ty(e) contrasts with the view that PPs act as predicates, which seems an appropriate analysis for prepositional phrases in copular sentences:

(62) Jamie is in the garden

A possible analysis for (62) would take either the preposition or the whole PP as being predicated of the subject, which, assuming that *Jamie* is of Ty(e), means that the PP appears as the functor:

```
(63)              • {Fo(in_the_garden'(jamie')), Ty(t)}
                                  /\
         • {Fo(jamie'), Ty(e)}    • {Fo(in_the_garden'), Ty(e -> t)}
```

Under this analysis, the copula is a mere placeholder for temporal information, and does not contribute to the establishment of predicate–argument structure, a view which is supported by languages which allow copula-less predication (e.g. Hebrew, Arabic, cf. Kempson, Edwards & Meyer-Viol 1998, Swinburne 1998), so that a uniform analysis can be achieved by assuming that the PP in copula constructions functions as a predicate.

A similar observation can be made about PPs functioning as adjuncts to NPs. I follow the general line proposed in Swinburne (1999) to analyse these cases of instances of the LINK operation, introduced in Chapter 1. The LINK operation establishes a relation between two (sub-)trees, with particular instructions or requirements which may vary from language to language[50]. Semantically (or, 'informationally'), a LINKed tree can be characterized as providing extra information for or about the expression it is LINKed to, i.e. the head. Thus, for example, the PP in (64) can be regarded as being LINKed to the subject:

(64) With his hat in his hand, the man left the room.

Following the analysis outlined in Swinburne (1999), the preposed PP is in a LINK relation to the subject, about which more information is given. The requirement imposed by the LINK relation is that at least one node between the two LINKed expressions be shared. For the example in (64), this would result in a structure like (65) (irrelevant details suppressed):

50 For example, in relative clauses as discussed in Chapter 1, the LINK operation may merely encode the requirement that there be a shared formula value between the head and the relative clause (as e.g. in Arabic), or it might provide a node with a copy of the formula value of the head with the requirement that the node by introduced into the relative clause (i.e. the LINKed tree) (as in English), cf. Kempson, Edwards & Meyer-Viol (1998).

(65)

```
                                  • {Tn(0) ?Ty(t)}
                                              / \
   • {Tn(00), Fo(the_man'), Ty(e)}     • {Tn(01), Fo(leave'(the_room')),
                          /                 Ty(e -> t)}
                       LINK
                        /
       • {Tn(00L0) ? Ty(p)}
                       /\
     • {Tn(00L00)     • {Tn(00L01),
         ?Ty(e)}       Fo(with_his_hat_in_his_hand'),
                       Ty(e -> p)}
```

There are a number of details in the tree in (65) which are worth commenting on. However, I am not concerned in the present context with LINK structures as such, but with the role of PPs in them. In particular, the (complex) PP in (65) acts as a predicate which takes the subject of the LINKed tree as an argument. Since the LINK operation requires that there be a shared value between the two trees, it is a copy of the formula value Fo(the_man') which can fill the subject node at Tn(00L00), so that the derivation results in a well formed structure. Semantically, the tree corresponds to the two related (i.e. LINKed) formulae, shown in (66):

(66) Fo(leave'(the_room')(the_man'))
Fo(with_his_hat_in_his_hand'(the_man')).

The second formula is, according to Swinburne (1999), of type Ty(p), a type specification for propositional structures which are not related to temporal location – in contrast to those of Ty(t), which are. Correspondingly, the PP here is of Ty(e -> p), so as to ensure that the shared formula value is incorporated into the tree and that a propositional structure results. As Swinburne demonstrates, this view results in a unified analysis of a number of cases of 'secondary predication', including PPs, participle clauses, and adjectival clauses, as in the following examples:

(67a) John left the office [with a cheerful smile].

(67b) The man, [sitting at the bar], had another gin.

(67c) Lucy, [angry at the crowd], stormed out of the cinema.

According to Swinburne, the bracketed strings in (67) are LINKed structures of Ty(p).

Swinburne's analysis differs from the analysis presented here with respect to the role of prepositions. While the LINK analysis assumes that prepositions are functors of Ty(e -> (e -> p)), the e* analysis assumes that prepositions have no type specification, but rather provide the tree structure for the introduction of optional Ty(e) expressions. However, the two analyses are concerned with different data, and distinct functions of prepositions in the building of semantic representations. While the two analyses taken together result in systematic ambiguity of the lexical specifications of (English) prepositions, they also provide a principled structural reflex of the different functions of PPs, where the dividing line is drawn between PPs which are "incorporated into the semantic unit the verb establishes" (Swinburne 1999: 217), that is, e*, and PPs which express a predication and, hence, build a LINKed tree.

One result of the two analyses is that attachment ambiguities receive distinct structural analyses as either involving e* or LINK:

(68) Bill saw the man with the telescope

(68') Bill ((saw ($_e$ the man))($_e$ with the telescope))

(68'') Bill (saw ($_e$ the man [$_{LINK}$ with the telescope]))

The ambiguity in (68) is represented as resulting from a possible interpretation of the PP as complement (68'), or as LINKed structure (68'').

In the light of the preceding discussion, the claim put forward in this thesis is thus that PPs can function as arguments to a verb, and that if so, they are of type Ty(e). Although the investigation of the full consequences which result from the division of labour between e* and LINK is a topic for future research, the two different analyses jointly provide the basis for an explanation of adjunct extraction, as shown in the next section.

5.4. Extraction

The analysis of VP and NP adjuncts as e* and LINK respectively results in an an account of the extraction patterns discussed in the preceding chapter. I discuss extraction of an NP, extraction of a PP, and the morphological registration of extraction in turn.

5.4.1. Extraction of NPs

A LINK relation indicates an island for extraction, while extraction is permitted along a path of functor–argument relations. Since by e*, NPs introduced by a preposition are associated at an argument node in the tree, they can be extracted. LINKed PPs, in contrast can not be extracted out of:

(69a) I went to the movies with Jane.
(69b) Who did you go to the movies with?

(70a) I liked the man with green hair.
(70b) ??What did you like the man with

Similarly, (71) can only be questioning a structure like (68'), but not (68"):

(71) What did you see the man with?

The answer *The telescope* as reply to (71) is only appropriate if the seeing was done with the telescope, but not if the man was with the telescope.

In LDSNL terms, this means that the unfixed constituent introduced by the *wh*-pronoun can be merged with a requirement TODO Ty(e) if this requirement is introduced by the preposition under e*, but not if there is an intervening LINK relation. The following parse stage shows the relevant tree configuration:

(72) *Tree for* "What did you see the man with?" *(e*)*

• {Tn(0), ?Ty(t)}
/ \
• {Tn(00), Fo($\mathbf{u}_{+addressee}$), Ty(e)} • {Tn(01), ?Ty(e -> t)}
/ \
• {Tn(010), Fo(the_man'), Ty(e)} • {Tn(011), ?Ty(e -> (e -> t))}
/ \
• {Tn(0110), +with, ?Ty(e), • {Tn(0111),
?Ty(e -> (e -> (e -> t)))}

{Tn(0*), Fo(WH), Ty(e)}
{Tn(0111*), Fo(see'), Ty(e* -> (e -> (e -> (e -> t))))}

The tree in (72) illustrates the derivation just before the unfixed nodes are merged. The Ty(e) node Tn(0110) has been built by lexical instructions from the entry for *with*. Note that now two unfixed nodes are in the tree, the unfixed

Ty(e) node from the pronoun and the unfixed e* node from the verb. By two applications of Merge, the nodes are associated at Tn(0110) and Tn(0111) respectively. The tree shows clearly why adjunct extraction works like argument extraction – the relevant tree configurations are identical.

In contrast, the unfixed Ty(e) node cannot be assigned an eventual location in a tree with LINKed NP adjunct:

(73) *Tree for* "What did you see the man with?" *(LINK)*

```
                         • {Tn(0), ?Ty(t)}
                                    / \
• {Tn(00), Fo(u_{+addressee}), Ty(e)}   • {Tn(01), ?Ty(e -> t)}
                                              /\
• {Tn(010), Fo(the_man'), Ty(e)}   • {Tn(011), ?Ty(e -> (e -> t))}
              /
           LINK
           /
           • {Tn(00L0), ?Ty(p)}
                   / \
• {Tn(00L00), ?Ty(e)}   • {Tn(00L01), ?Ty(e -> p)}
                             / \
  • {Tn(00L010), ?Ty(e)}   • {Tn(00L011), Fo(with'),
                              Ty(e -> (e -> p))}

{Tn(0*), Fo(WH), Ty(e)}
{Tn(011*), Fo(see'), Ty(e* -> (e -> (e -> t)))}
```

The tree in (73) results in an incomplete, and hence disallowed, derivation since the unfixed Ty(e) node can not be introduced into the tree. The underspecified tree node cannot be resolved to hold at Tn(00L010) since the Kleene star operation over tree node addresses is defined only over any number of 0's and 1's, but not over any address involving an L (i.e. a LINK relation). Since there is no requirement TODO Ty(e) in the matrix tree, all putative applications of Merge lead to inconsistency, and the derivation is aborted.

5.4.2. Extraction of PPs

The e* analysis developed so far provides, as shown in the preceding section, a structural explanation for NP extraction out of VP adjunct position with stranded preposition, such as for example in (74):

(74) Who did you go to the movies with?

However, for questions with fronted PPs such as (75), the analysis has to be slightly amended:

(75) With whom did you go to the movies?

Examples like (75) should be subject to the same analysis as standard object extraction and NP extraction as discussed in the last section, namely as involving an unfixed Ty(e) node which is integrated into the tree by e* and Merge, so that before Merge the following parse state obtains:

(76) *Tree for* "With whom did you go to the movies?

```
                              • {Tn(0), ?Ty(t)}
                                      / \
• {Tn(00), Fo(u+addressee), Ty(e)}   • {Tn(01), ? Ty(e -> t)}
                                          /\
• {Tn(010), Fo(to_the_movies'),       • {Tn(011),
                          Ty(e)}         ?Ty(e -> (e -> t))}

{Tn(0*), Fo(with_WH), Ty(e)}
{Tn(011*), Fo(go'), Ty(e* -> (e -> (e -> t))))}
```

The challenge PP extraction poses is that the lexical instructions of prepositions as currently formulated state that the condition for their introduction into the tree is that there be a requirement TODO Ty(e* –> (e –> t)). In question formation, however, the preposition is found at the outset of the parse, where the requirement TODO Ty(t) holds. In the light of the tree in (76), the lexical entry for prepositions has to be modified such that they can be introduced at the beginning of the parse, and that if they are, they are integrated into the verb phrase. This can be achieved by a disjunctive lexical entry for prepositions, which is, however, activated by a uniform condition[51]:

---

51 This uniform condition might turn out to be important since if the LINK analysis discussed above is integrated into the system, lexical statements for prepositions need to include more information, including a statement to the effect that LINKed prepositions have type information. The way the entry is built here ensures that everything relevant for e* will be found under one condition.

(77) *Schematic Disjunctive Lexical Entry for Prepositions*

IF	?Ty(e* –> t)	
THEN	IF	?Ty(t)
		put(? <d*> +prep, Ty(e))
		make(<d*>), put(+prep, ?Ty(e))
	ELSE	put(?<d_1> Ty(e –> (e –> t))),
		make(<d_0>), put(+prep, ?Ty(e))
ELSE	abort	

The lexical entry in (77) states firstly the condition for the introduction of prepositions, where the definition of e* is finally fully exploited, since now possible instantiations include a simple Ty(t). However, two cases are distinguished by the two clauses under THEN. The first case is where the preposition is introduced directly under the root node, that is when a requirement TODO Ty(t) holds, as is the case in questions. If this condition is met, two actions are performed. The root node is annotated with a modal requirement that down of the root node a Ty(e) expression with a prepositional annotation holds. This statement ensures in conjunction with Prediction that the PP can eventually be introduced into the tree. The second clause states that an unfixed node be built where an expression of Ty(e) is required. At this node, the interrogative pronoun, a Ty(e) expression is introduced. The resulting expression at the unfixed node meets the requirement which the first clause puts at the root node, so that it can later be introduced into the tree. The ELSE clause is the ordinary case where the preposition builds a node for an optional Ty(e) expression. This clause remains unchanged. The relevant parse states are given below:

(78a) • {Tn(0), ? Ty(t)}

The initial state of a parse provides the condition for the introduction of a preposition. This results in (78b):

(78b) • {Tn(0), ?Ty(t), ?<d*> +prep, Ty(e))}

{Tn(0*), +prep ?Ty(e))}

The following *w h*-pronoun fulfills the requirement at Tn(0*) and annotates the root node with the question feature +Q (as standardly assumed):

```
(78c)                    • {Tn(0), +Q, ?Ty(t), ?<d*> +prep, Ty(e))}

        {Tn(0*), +prep, Fo(prep_WH), Ty(e))}
```

The following steps proceed in a standard fashion to result in a tree like (78d), which is identical, in the relevant respects, to the tree in (76) above[52]:

```
(78d)
                          • {Tn(0), +Q, ?Ty(t), ?<d*> +prep, Ty(e))}
                                 / \
                 • {Tn(00), Ty(e)}  • {Tn(01), ?Ty(e -> t)}
                                        /\
                 • {Tn(010), Ty(e)}   • {Tn(011), ?Ty(e -> (e -> t))}

        {Tn(0*), +prep, Fo(prep_WH), Ty(e))}
        {Tn(011*), Ty(e* -> (e -> (e -> t))))}
```

The next step in the derivation is the application of Prediction, which results in the building of a node with the requirement specified in the root node:

```
(78e)
                          • {Tn(0), +Q, ?Ty(t), ?<d*> +prep, Ty(e))}
                                 / \
                 • {Tn(00), Ty(e)}  • {Tn(01), ?Ty(e -> t)}
                                        /\
                 • {Tn(010), Ty(e)}    • {Tn(011), ?Ty(e -> (e -> t))}
                                       /
            • {Tn(0110), ? +prep, Ty(e))}

        {Tn(0*), +prep, Fo(prep_WH), Ty(e))}
        {Tn(011*), Ty(e* -> (e -> (e -> t))))}
```

At this stage, Merge applies to Tn(n0*) and Tn(0110) so that the underspecified Ty(e) node is introduced into the tree:

52 That is, without formula values and now including the changes resulting from the new lexical specification for prepositions.

(78f)

```
                    • {Tn(0), +Q, ?Ty(t), ?<d*> +prep, Ty(e))}
                         / \
     • {Tn(00), Ty(e)}  • {Tn(01), ?Ty(e -> t)}
                             /\
      • {Tn(010), Ty(e)}    • {Tn(011), ?Ty(e -> (e -> t))}
                               /
        • {Tn(0110), +prep,
     Fo(prep_WH), Ty(e))}

{Tn(011*), Ty(e* -> (e -> (e -> t))))}
```

With this step, the requirement introduced by the preposition at Tn(0) is fulfilled, since a Ty(e) expression with a +prep feature holds down of Tn(0). The derivation can now be completed by Prediction and Merge, and eventually by Completion.

With the disjunctive lexical defintition of prepositions as given in (77) both question patterns of PP adjuncts can now be derived.

5.4.3. Registration of Extraction Paths

A final point with respect to extraction to be addressed is the explicit syntactic or morphological marking of argument and adjunct extraction as discussed in the preceding chapter. Given the important role of lexical instructions for the analysis presented here, cross-linguistic differences are expected to reside to some extent in the characterization of prepositions, case, and NPs, as well as general rules of tree building. Thus, no full analysis of the languages discussed by Hukari & Levine (1994, 1995) showing marking of adjunct extraction can be produced here. Rather, I just show how the symmetrical behaviour can be stated, irrespective of the actual analysis of the morphological or syntactic reflex.

Consider again the case of the Irish complementizers (McCloskey 1979, quoted from Hukari & Levine 1995: 206):

(79a) *I mBetlehem* *aL* *dúirt na targaireachtaí*
[in Bethlehem]$_j$ COMP said the prophecies

aL *béarfaí* *an Slánaitheoir*
COMP would-be-born the Saviour e_j

'It was in Bethlehem that the prophecies said that the Saviour would be born'

(79b) *Cén uair* *aL tháing siad na bhaile*
[which time]$_j$ COMP came they home e_j

As seen in the examples, the complementizer is *aL* if an argument or an adjunct is extracted. In LDSNL terms, the difference between the two environments is that in (79) an unfixed Ty(e) node (and an appropriate +Q feature) has been introduced into the tree, as opposed to the non-extraction environment, where all Ty(e) nodes are fixed. A possible lexical analysis of the complementizer *aL* in Irish could express this as follows:

(80) *Lexical Entry for Irish Complementizer* aL

IF	?Ty(t –> (e* –> (e –> t)))	
THEN	IF	exist({Tn(n*), Ty(e)})
	THEN	make(<d_0>), put(?Ty(t), ?<d*> Ty(e)))
ELSE	abort	

The entry in (79) assumes that *aL* is a context sensitive allomorph of an 'abstract' lexical complementizer. For lexical access, this means that the parsing of one form necessarily accesses the other. The two forms are then reflexes of the different clauses in the lexical entry. For the case of *aL*, the entry states the condition for the existence of an unfixed node, which does not need to be fulfilled for *goN*, which is used in sentences without extraction. In the main condition, I have included a tentative type for verbs taking sentential complements which allow for the introduction of Ty(e) expressions before the introduction of the sentential complement. However, a full treatment of this question is outside the scope of the thesis, and thus the formulation in the IF clause is only tentative. What is relevant in (79) is that the condition stated for the introduction of *aL* in the lexical entry is sensitive to unfixed Ty(e) expressions, so that *aL* is used for both extracted arguments and extracted adjuncts.

The brief consideration of Irish here does not purport to be a serious analysis. The only relevant point is that the symmetry of arguments and adjuncts follows in the e* analysis from the fact that both questioned arguments and questioned adjuncts are assigned to an unfixed node of Ty(e), and that this fact can be exploited in stating whatever language particular reflex it has. Furthermore, the characterization of verbs as involving an unfixed, but conditional, type ensures that the two kinds of underspecified nodes can always be distinguished.

The evidence from extraction thus confirms the e* analysis of verbal underspecification, which provides a means to introduce optional Ty(e) expressions into the verb phrase as arguments, so that optional and obligatory Ty(e) expressions can be seen to behave alike with respect to extraction.

5.5. e* and Unfixed Verbs: German

In the discussion so far, a number of points have been discussed mainly with English in mind. In this section I offer a short discussion of German and show how the e* analysis works in that language. As briefly pointed out above, one reason for introducing e* predicates from the lexicon rather than by general rule is that a general rule of e* Adjunction might be needed more urgently for an analysis of verb movement. In this section I discuss this point with reference to German. For the purposes of this thesis I assume that German is rigidly verb-second/verb-final, that is that the tensed verb, including tensed lexical verbs, is found either after the first constituent of the clause, or in final position[53]. Furthermore, I ignore issues involving auxiliary and modal verbs, including cross-modal dependencies[54]. The point of interest here is then how e* predicates interact with the generally unfixed nature of verbs in German.

As in English, PPs can freely be added into the VP in German, both in main and in subordinate clauses:

(80a) *Frank sang*
Frank sang
'Frank was singing'

(80b) *Frank sang Arien*
Frank sang arias
'Frank was singing arias'

53 Both of these assumptions are necessary in view of the examples below:

(i) *Den Peter den habe ich schon ewig nicht mehr gesehen.*
the-ACC P the-ACC have I already forever not anymore seen
'Peter I haven't seen in ages'

(ii) *Petra glaubt daß Sonja gesagt hat daß Holland morgen gewinnt.*
P believes that S said has that Holland tomorrow win
'Petra believes that Sonja said that Holland will win tomorrow'

In (i), two NPs precede the tensed verb in a main clause, while in (ii) the sentential complement of *sagen*, 'say', follows the verb. However, I go not into these details here.

54 Cf. e.g. Hinrichs & Nakazawa (1994).

(80c) *Frank sang in der Wanne*
Frank sang in the.DAT tub.DAT
'Frank was singing in the bath-tub'

(80d) *Frank sang aus vollem Halse Arien in der Wanne*
F. sang out full.DAT throat.DAT arias in the.DAT tub.DAT
'Frank was singing arias at the top of his voice in the bath-tub'

(80e) *... daß Frank aus vollem Halse Arien in der Wanne sang*
that F. out full.DAT throat.DAT arias in the.DAT tub.DAT sang
'that Frank sang arias at the top of his voice in the bath-tub'

The examples show *singen*, 'sing', with different arguments and adjuncts; the intransitive use (80a), transitive use with optional argument (80b), with a locative adjunct (80c), with optional argument, locative and manner adverb in main (80d) and subordinate clause (80e). The order of the VP constituents is, at least syntactically, unfixed:

(81a) *Frank sang Arien aus vollem Halse in der Wanne*
F. sang arias out full.DAT throat.DAT in the.DAT tub.DAT
'Frank was singing arias at the top of his voice in the bath-tub'

(81b) *Frank sang in der Wanne Arien aus vollem Halse*
F. sang in the.DAT tub.DAT arias out full.DAT throat.DAT
'Frank was singing arias at the top of his voice in the bath-tub'

(81c) *... daß Frank in der Wanne aus vollem Halse Arien sang*
that F. in the.DAT tub.DAT out full.DAT throat.DAT arias sung has
'that Frank sang arias at the top of his voice in the bath-tub'

Furthermore, the position of the tensed verb varies between second and final position in main and subordinate clause.

Given the data in (80) and (81), I assume that the e* analysis of verbal underspecification holds for German, including the analysis of optional arguments involving disjunctive lexical entries, and the general requirement that Ty(e) arguments needs to be licensed to be introduced into the tree. However, in contrast to English, the structure building process cannot be characterized as proceeding to a large extent from lexical information from the verb, but has to be achieved in a way similar to how I have characterized the introduction of optional arguments in English, namely by structure building operations from prepositions and case. From this perspective, the final position of the verb can be taken to be basic, so that subcategorization requirements can be checked against already established tree structure.

Consider for example a parse for a simplified version of (81c):

(82) *... daß Frank in der Wanne Arien sang*
that F. in the.DAT tub.DAT arias sung has
'that Frank sang arias in the bath-tub'

The corresponding tree just before the introduction of the verb is given in (83):

(83a) *Tree for "...* Frank in der Wanne Arien

```
                         • {Tn(0), ?Ty(t)}
                                  / \
• {Tn(00), Fo(frank'), Ty(e)}   • {Tn(01), ?Ty(e -> t)}
                                          / \
   • {Tn(010), Fo(in_der_wanne'),      •{Tn(011),
                          Ty(e)}       ?Ty(e -> (e -> t))}
                                          / \
        • {Tn(0110), Fo(arien'), Ty(e)}   • {Tn(0111),
                                          ?Ty(e -> (e -> (e -> t)))}
```

In order to ensure a particular order relation on the introduction of Ty(e) expressions, an explicit characterization of the function of prepositions and case in German has to be given, as well as an addtional notion of locally unfixed Ty(e) nodes. Although these are interesting questions, I assume that Ty(e) expressions are introduced into the tree in the order in which they appear, since I am here interested in the introduction of the verb. The next step in the derivation in (83) is the scanning of the verb. On the assumption that *singen* is transitive in the relevant reading here, it specifies the obligatory presence of the subject and the object, and is thus of Ty(e* -> (e -> (e -> t))). The important point with respect to this final position is that the verb does not have to be assigned to an unfixed position – when the verb is introduced in final position, no more Ty(e) expressions can be introduced, all expressions which determine the predicate's eventual valence are already in the tree. The derivation thus proceeds immediately with the association of the DU introduced by the verb at Tn(0111):

(83b) *Tree for* "... Frank in der Wanne Arien sang

```
                       • {Tn(0), ?Ty(t)}
                              / \
• {Tn(00), Fo(frank'), Ty(e)}   • {Tn(01), ?Ty(e -> t)}
                                         / \
   • {Tn(010), Fo(in_der_wanne'),     •{Tn(011),
                           Ty(e)}      ?Ty(e -> (e -> t))}
                                          / \
        • {Tn(0110), Fo(arien'), Ty(e)}  • {Tn(0111)
                                          Fo(singen'),
                                          Ty(e* -> (e -> (e -> t)))
                                          ?Ty(e -> (e -> (e -> t)))}
```

The tree in (83b) shows this step. The information at Tn(0111) is consistent and the requirement can be checked by Thinning. In German, then, the verb when introduced in final position is not assigned to an unfixed position. This is expressed in the following lexical entry:

(84) *Lexical Entry for* singen *(first version)*

```
IF      ?Ty(e* -> (e -> (e -> t)))
THEN    put(Fo(singen'), Ty(e* -> (e -> (e -> t)))),
        go(<u*> +ACC), go(<u*> +NOM),
        go(<u*> ?Ty(t)).
ELSE    abort
```

The requirement in (84) indicates that transitive *singen* cannot be introduced into the tree unless at least two Ty(e) expressions have been introduced into the tree. This is important, since, as already said, no further Ty(e) expressions can be introduced at this stage, so all required Ty(e) expressions have to be present. If this condition is fulfilled, the formula and type values of the verb are introduced at the node where the requirement holds. The go predicates have two functions. The first two of them ensure that the Ty(e) expressions are indeed the ones required by the verb, and not optionally introduced ones. The instructions cause the pointer to go up the tree to check for the case feature +ACC and +NOM, where the object is lower than the subject. The third go predicate causes the pointer to go to the mother node Tn(0) where TODO Ty(t) holds. This is to ensure that no further expressions can be introduced into the tree. The combined effect of this definition is that the verb is introduced directly at a fixed position in the tree, and that subcategorization requirements are checked, since they have to be met at this stage.

The lexical definition given so far has now to be extended to cover V2 cases. It is for the analysis of V2 that a general e* Adjunction rule is needed. The first additional observation in this respect is that the intitial position in German is not reserved for subjects. Rather, in principle any constituent may be introduced into the parse intitially:

(85a) *Frank sang aus vollem Halse Arien in der Wanne*
F. sang out full.DAT throat.DAT arias in the.DAT tub.DAT
'Frank was singing arias at the top of his voice in the bath-tub'

(85b) *In der Wanne sang Frank Arien aus vollem Halse*

(85c) *Arien sang Frank in der Wanne aus vollem Halse*

(85d) *Aus vollem Halse sang Frank Arien in der Wanne*

The data in (85) show that all constituents can be found before the verb. In LDSNL terms this means that the initial position is a locally unfixed position which will be fixed only at some later stage during the parse. This in turn means that the verb in V2 is introduced at a stage where TODO Ty(t) holds. The building of the initial unfixed position can be characterized by a general adjunction rule[55]:

(86) *e Adjunction*

$$\frac{\{_n \ldots ?\,Ty(t)\}}{\{_n \ldots ?\,Ty(t), <d^*>\, ?Ty(e)\}}$$

The rule e Introduction licenses the introduction of an unfixed node with the requirement TODO Ty(e) at the root node. The eventual position of the node depends then on case or prepositional information. On the assumption that the introduction of the initial Ty(e) expression always involves e Adjunction, the condition for the introduction of the verb in V2 position can be stated as a requirement TODO Ty(t):

55 Cf. the rule Star Adjunction, introduced in Chapter 1.

(87) *Lexical Entry for* singen *(final version)*

```
IF      ?Ty(e* -> t)
THEN    IF      ?Ty(t)
          THEN  make(<d*>),
                put(Fo(singen'), Ty(e* -> (e -> (e -> t)),
                      <u*> +ACC, <u*> +NOM)
                go(<u*> ?Ty(t))
        OR      ?Ty(e* -> (e -> (e -> t)))
          THEN  put(Fo(singen'), Ty(e* -> (e -> (e -> t)))),
                go(<u*> +ACC), go(<u*> +NOM),
                go(<u*> ?Ty(t)).
ELSE    abort
```

The new part of the entry, the first disjunct of the main THEN statement, licenses the building of an unfixed node for the verb. The go statements of the verb-final entry are now modal requirements, ensuring that both subject and object are found above the verb's eventual position in the tree. The second part of the entry remains unchanged. The new entry licenses the following step in the derivation of a V2 verb:

(88) *Tree for "In der Wanne sang*

• {Tn(0) ? Ty(t)}

{Tn(0*), Fo(in_der_wanne'), Ty(e)}
{Tn(0*), Fo(singen'), Ty(e* -> (e -> (e -> t)))}

The remaining steps in the derivation proceed standardly, that is, the tree is developed similar to derivations with final verb, where the only difference is that the verb has already been assigned to an unfixed position. From the perspective of the verb, the derivation proceeds exactly as in English in that Merge applies after all Ty(e) expressions have been scanned. I thus do not give the remaining steps of the derivation.

What this brief consideration has shown is that e* can be used for languages with verb movement, under an analysis which specifies different conditions of introduction in a disjunctive lexical entry. There are two important points to note for the e* analysis. The first is that it is not always the case that underspecified types and unfixed location go together. Although they do go together English and in V2 in German, the final verb is crucially introduced into the tree directly. This is the reason for introducing e* predicates from the lexicon, rather than by general rule, since this treatment ensures that the two kinds of underspecification can be separated. The second point to note is that incremental interpretation of e* predicates doesn't work for final verbs in German, since in these cases, the predicate is introduced directly, so that no application of Merge is necessary. Rather, the underspecification of underspecified predicates is resolved simply by Thinning. In view of this fact, the interpretation of e* predicates can better be analysed as a in-one-go process. However, for the next chapter, where an incremental semantics for e* is developed, I do in fact assume that e* predicates can be incrementally interpreted, and that the rule e* Partial Resolution is operative. The assumption is dropped in Chapter 5, so that I do not provide an incremental resolution for final verbs in German here[56].

6. Conclusion

In this chapter I have introduced the main proposal made in this thesis, namely that the subcategorization information provided by verbs is underspecified in the sense that the type value provided by verbs explicitly licenses the optional introduction of arguments. I have provided a formal statement of this observation – e* – and have shown how this underspecified type interacts with the overall LDSNL system. Since e* employs resources independently provided by the system, it can be used without major revisions. The formalization models successfully both the optionality of verb phrase adjuncts, and their behaviour as arguments once they are introduced into the tree. The building of verb phrases can, under this proposal, be characterized as the interplay between lexically provided information and more freely available general processes. In the next two chapters, I turn to the question of how verbs with underspecified types can be semantically interpreted. Chapter 4 provides a formalization of incremental semantic interpretation of underpecified verbs, where each introduction of a Ty(e) expression into the tree is matched by a corresponding semantic operation on the formula value (i.e. the lambda expression) introduced by the verb. However, some problems with this analysis are noted, so that in Chapter 5 I explore an alternative to this semantic treatment, which relates the interpretation of underspecified verbs to the pragmatic process of concept formation.

---

56 One possible way to do this is to state incrementality over Completion, since by the time Completion applies all predicates have a fixed arity and the application of modus ponens over formula values proceeds in a step-by-step fashion.

Chapter 4

Semantic Interpretation for Underspecified Verbs

1. Introduction

In the preceding chapter, I have proposed a formalization of verbal underspecification as an analysis of optional Ty(e) expressions in the verb phrase. I have shown that certain symmetries between arguments and adjuncts can be expressed in the system by an underspecified verbal type, e*. The formalization captures the fact that the main difference between arguments and adjuncts is the difference between obligatoriness of the former and optionality of the latter. The proposal furthermore provides a unified analysis of the role of NPs and PPs in verb modification, as well as expressing the syntactic constituency of PPs. The formulation of verbal underspecification as e* has been shown to be closer to general LDSNL assumptions than the possible alternative treatment as expressed by the rule of Adjunction. e* is thus a natural extension of the framework which exploits independent resources, working on the inherent dynamics of the system. However, the discussion so far has mainly been concerned with the syntactic properties of underspecified verbs and their role in the establishment of tree structure. Thus far, I have not discussed semantic aspects of e* in detail, which will be the subject matter of this chapter. The first section, I discuss three analyses of adjunction from the formal/categorial grammar tradition which complement the syntactic distinction between Adjunction and e* of the last chapter; Dowty's (1979) intensional semantics, McConnell-Ginet's (1982) extensional semantics, and Minimal Recursion Semantics (Copestake, Flickinger & Sag 1997). This chapter thus also serves to show how the argument developed in this thesis relates to proposals reported in the literature. After discussing the three analyses presented, I develop semantic rules for underspecified verbs, based on the work of McConnell-Ginet, which provide a means to assign model-theoretic interpretation to verbs with an underspecified type. However, the evaluation of the rules leads, together with LDSNL assumptions about the interpretation of natural language expressions, to the rejection of this analysis, and provides the transition to the next chapter, where the role of underspecified verbs in relation to conceptual structure is discussed.

2. Adjuncts as Functors or as Arguments

In the discussion of the two alternative formalizations of adjunction in LDSNL in the last chapter – Adjunction and e* – I have omitted a discussion of the semantics of these two approaches, since the discussion was concerned with the dynamics of tree building and an evaluation of the two proposals with respect to central LDSNL assumptions. The focus of the chapter also meant relevant proposals in the literature were not discussed in depth. By turning to the semantics of e* in this chapter, I discuss three proposals from the Formal/Categorial Grammar tradition, which are concerned with effectively the same problem, albeit from a different perspective, namely the correct representation of adjuncts, and of adverbials more generally. As will be seen, the semantics of adjuncts developed by Dowty (1979) can not be simply extended to the Adjunction rule of the last chapter, despite superficial similarities (2.1), while, in contrast, the proposal developed by McConnell-Ginet (1982) can be adapted to provide a semantic rule for e* (2.2), which is presented in section 3. The last proposal discussed, Minimal Recursion Semantics (Copestake et al. 1997), is shown to provide a useful technical tool for developing a semantics in LDSNL, but it is not further developed for conceptual reasons.

2.1. Adjuncts as Functors

The Adjunction rule discussed in the last chapter follows the treatment developed in Dowty (1979) (and in fact in Montague 1973), who combines Montague Grammar with the lexical decomposition approach advocated within Generative Semantics (GS) (cf. e.g. Lakoff 1971). Following Montague, Dowty develops a possible world semantics for a type-logical system, and combines this with the 'abstract' underlying predicates CAUSE and BECOME introduced within GS. These predicates are, together with the temporal predicates such as PRES, FUT, PAST, defined in a temporal possible-world semantics. The two main underlying assumptions in Dowty's work, that truth -theoretic semantic meaning is assigned directly to natural language expressions, and that words can be decomposed into smaller underlying meaning-bearing units, are not shared by LDSNL. Both these points are further discussed in the final Section 4 of this chapter. In this section, I merely want to show that Dowty's semantic rules if transferred to LDSNL create more syntactic problems for Adjunction, and that his semantic rules can likewise not easily be transferred to LDSNL. The main point of this discussion is to show that there are no independent semantic reasons for motivating Adjunction, and that given an appropriate formulation of the semantics of e*, the latter account is to be preferred.

2.1.1. IV/IV and TV/TV

In the light of examples like (1) and (2), Dowty (1979: 207/208) proposes to distinguish between intransitive verb and transitive verb modifiers:

(1a) John walked.
(1b) John walked to Chicago.

(2a) John moved the rock.
(2b) John moved the rock to the fence.

The PPs in (1) and (2) are similar in that they turn an 'activity' verb into an 'accomplishment' verb, that is, in Dowty's system, they add a BECOME predicate to the verb. However, (1b) entails that the referent of the subject ends up in Chicago, while (2b) entails that the referent of the object is at the fence, which can be expressed with a CAUSE predicate. Thus, there are two prepositions *to*; the first denotes a function from intransitive verbs to intransitive verbs (IV/IV), the second denotes a function from transitive verbs to transitive verbs (TV/TV). The first of these is illustrated in the analysis tree below (Dowty 1979: 211)[57]:

(3a) *Analysis Tree for* "John walks to Chicago"

```
              John walks to Chicago, t
                 /           \
           John, T            walk to Chicago, IV
                                    /  \
               to Chicago, IV/IV    walk, IV
                   /  \
       to, (IV/IV)/T   Chicago, T
```

57 I have omitted references to the syntactic rules licensing steps of combination which are given in the original. Though obviously related, the Montagovian type system is different from the types employed in LDSNL, as briefly pointed out above (Chapter 2, Section 2.1.); here approximate correspondences are t = Ty(t), T = Ty(e), IV = Ty(e -> t), and (IV/IV)/T = Ty(e -> ((e -> t) -> (e -> t))).

The most interesting step to note here is the combination of the verb and the PP, where the PP acts as functor which takes the verb as argument. The preposition is correspondingly typed. The semantics for the preposition, a complex lambda expression, results after appropriate conversion in the translation in (3b):

(3b) *Translation for* "John walks to Chicago"

walk'(j) & BECOME [be-at'(j, c)]

Thus here, the preposition introduces the BECOME predicate (which is interpreted as meaning intuitively a temporal 'before–after' sequence) and the locational predicate be-at', which is not further analysed, although Dowty indicates ways to provide a more detailed analysis by employing a LOC function assigning positions to individuals in a (every) model. The sentence is interpreted as a conjunction of two predicates, where walk' remains unary, and the relation between the referents of subject and object is expressed by be-at' (hence the preposition is 'transitive').

The second *to* is illustrated in (4) (Dowty 1979: 212):

(4a) *Analysis Tree for* "John pushes a rock to the fence"

```
John pushes a rock to the fence, t
          /     \
John, T         push a rock to the fence, IV
                         /     \
              a rock, T        push to the fence, TV
              |                             / \
           rock, CN         to the fence,      push, TV
                               TV/TV
                               /     \
              to, (TV/TV)/T         the fence, T
                                          |
                                    fence, CN
```

In (4a), in contrast to (3a), the preposition combines with its object to form the transitive verb modifier *to the fence*. There is the technical problem that when the 'transitive verb' *push to the fence* combines with the object of *push*, that is, *a rock*, the object has to be 'inserted' between the verb and the PP, which have already combined – the solution being some kind of 'wrap' operation without semantic content. While this is not problematic in Montague Grammar (cf. McGee Wood 1993, Morrill 1994), such a mechanism would have to be defined in LDSNL if the system were to incorporate Dowty's semantics. The translation corresponding to (4a) is then:

(4b) *Translation for* "John pushes a rock to the fence"

$\exists x$ [rock'(x) & $\exists y$[$\forall z$ [fence'(z) <–> y = z] & push'$_*$(j, x) CAUSE BECOME [be-at'(x, y)]]]

As (4b) shows, the transitive modifier preposition introduces, in addition to BECOME, a CAUSE predicate into the translation, indicating that it is the object which 'moves', and not the subject[58].

The two translations for prepositions (*to* is just an example) as IV/IV and TV/TV lead to a systematic ambiguity in the lexicon in that there are (at least) two entries for each preposition. Dowty proposes a translation rule, which translates intransitive modifiers into transitive modifiers, so that prepositions only need to be listed as IV/IV. This means that the semantic operation associated with TV/TV can be defined on the operation IV/IV[59]. But this operation cannot be integrated into LDSNL as it stands, since, as pointed out in Chapter 3, lexically established types cannot be changed. The solution for this problem sketched in Chapter 3 was to 'underspecify' the types of modifiers by using variables over types such as Ty(e –> (X –> X) for prepositions. But given Dowty's semantics, this type underspecification really translates into a disjunction for every preposition such that the Adjunction rule has to encode that if X = Ty(e –> t), apply semantic rule A, but if X = Ty(e –> (e –> t), apply semantic rule B. That is, the ambiguity is not resolved[60].

In order to give a consistent interpretation of PPs as modifiers, Dowty translates verbs which subcategorize for a PP such as *put, set, lay* as requiring a modifier as argument. That is, *put* is of category TV/(TV/TV), requiring a transitive verb modifier to yield a transitive verb, corresponding to an LDSNL type Ty(((e –> t) –> (e –> t)) –> (e –> (e –> t))). Again, this is not a possible

58 The clause $\forall z$ [fence'(z) <–> y = z] indicates that *the fence* is definite, the notation push'$_*$ means that *push* translates into an 'extensional' predicate, in contrast to for example *seek*. Both points are irrelevant here.

59 I do not reproduce the translation rule here, nor for that matter the actual translation rules for prepositions. For the present discussion, the difference between IV/IV and TV/TV as the absence or presence of the CAUSE predicate is all that is needed. The actual rules involve the intensional interpretation of all expressions and are without a discussion of the relevant interpretations not very illuminating. However, the issue of intensionality is not really relevant here and will only briefly be discussed in relation to McConnell-Ginet's criticism of Dowty in the next section.

60 In view of the lexical entries discussed so far, and of those to be introduced in Chapter 6, this is in fact not a very forceful criticism.

expression under the interpretation of Type in LDSNL. The background of this discussion about possible types is that, as briefly discussed in Chapter 2, while Categorial Grammar treats types as an inference system, allowing, for example, for transitivity of types and function composition, LDSNL types are interpreted much more restrictedly as transparently encoding lexically determined syntactic combinatory properties. It is this difference between the two systems which makes it difficult to integrate Dowty's work on PPs into LDSNL. The comparatively weak (but sufficient for present purposes) claim made here is thus that despite appearances, there is no straightforward semantic motivation for an Adjunction rule in LDSNL, on top of the syntactic problems the rule brings with respect to the tree building noted in the last chapter.

2.1.2. Some Problems Noted by Dowty

Dowty himself notes that the analysis of PPs as either IV/IV or TV/TV is not without problems. Since TV/TV results from a translation rule applying to IV/IV, all PPs can function as IV/IV, since the translation rule is optional. For example, the PP in (5) can apply as IV/IV to the 'VP':

(5) John threw the letter into the wastebasket.

The preferred interpretation here is that the PP is a TV/TV so that the letter ends up to be in the wastebasket. However, the PP might equally well function as IV/IV in which case it is John who comes to be in the wastebasket (1979: 233). These two readings are always possible within the system, but one reading is 'highly unlikely' for pragmatic reasons. A slightly different situation arises with (6):

(6) John drove his car to Chicago.

In (6), both interpretations seem to be possible (or, maybe, required) since both John and his car end up in Chicago. Dowty proposes that sentences like (6) "may well be syntactically (and semantically) ambiguous, though the two readings are indiscernable for pragmatic reasons" (1979: 209). The assumption here is that pragmatics chooses between different readings generated by the syntax/semantics. Given, however, that the semantics here is largely set up to capture exactly entailments like the ones relevant in (5) and (6) (by introduction of 'CAUSE to be in/at...'), the division of labour between semantics and pragmatics seems to be not clear. In addition, it is not clear either how the role of the predicate, that is, the verb, in these entailment relations can be expressed, since the analysis assigns semantic operations of PP modification to the preposition. For example, the contrast between (6) and (7) seems to result from the choice of the predicate, and not from the PP:

(7) John wrote a letter to the editor.

That is, the particular pragmatic knowledge about who and what ends up where involved in these cases includes knowledge about driving and writing, that is, about concepts addressed by the verbs, yet by treating PPs as functors, this observation is not expressed directly.

2.1.3. Concluding Discussion

The point of discussing Dowty's analysis of prepositions and PPs was to show that his semantic rules can not straightforwardly be incorporated into LDSNL, despite the superficial similarity with the Adjunction rule. It turns out that the only similarity is that in both systems, PPs are treated as functors which take verbs as arguments. However, Dowty employs a typing system which is very different from the LDSNL types, and proposes a semantic analysis which, in addition to being completely intensional, employs predicates like BECOME and CAUSE resulting from 'lexical decomposition'. The reasons for not introducing these notions into LDSNL have to do with assumptions about the relation between natural languages strings and truth conditional content, which for Dowty is direct, while in LDSNL, the relation is mediated by an intermediate level of representation, over which semantic interpretation is assigned. Since under this view, the contribution of lexical items is to address a mental concept, lexical decomposition is not a natural solution from the perspective adopted here. This point is discussed further in the last section of this chapter, but for the moment, it suffices to point out that providing a semantics for Adjunction along the lines discussed here would have unwanted consequences for the LDSNL system, so that there is no 'independent' semantic motivation for Adjunction.

2.2. Adjuncts as Arguments: McConnell-Ginet (1982)

An alternative semantics for adjuncts as been proposed by McConnell-Ginet (1982). In fact, some of the syntactic arguments for preferring e* over Adjunction go back to this work. However, although developed as an alternative to Dowty, McConnell-Ginet's account is mainly concerned with true lexical adverbs such as *rudely, quickly,* and leaves certain questions about the role of NP and PP adjuncts open. In my discussion of McConnell-Ginet's analysis I concentrate on those aspects which are relevant for e* and largely ignore her main arguments relating to adverbs in general. Since her syntactic arguments have been partly presented already, I review them only briefly here. The intuition behind and formalization of her proposal for the semantics of adjuncts, in contrast, will be presented more extensively.

McConnell-Ginet's main claim is that a treatment of adjuncts as functors (e.g. of Ty((e –> t) –> (e –> t)) as in Adjunction) has two serious problems. It fails to express entailments like the one from (8a) to (8b), and it forces an analysis in terms of possible worlds, which, according to McConnell-Ginet, seems superfluous in these cases (1982: 165):

(8a) Linda spoke to Marcia.

(8b) Linda spoke.

Intuitively, the PP in (8a) specifies the predicate – not only did Linda speak, but she also addressed Marcia in doing so. Conversely, speaking to someone entails speaking, so (8b) is entailed by (8a). To illustrate that by Adjunction, this relation is not expressed, McConnell-Ginet discusses the analysis of the adverb *quickly* as functor. If *quickly* is combined with an intransitive predicate like *walk* as an argument to result in an (another) intransitive predicate, the corresponding ('standard') semantic operation is the mapping from a function from individuals to truth values to another function from individuals to truth values[61]. However, there is no explicit formal statement in this semantic operation that there is a relation between the first set of individuals (the walkers) and the second set of individuals (the quickly_walkers): "Given the semantics proposed, there is no reason to conclude that talking quickly is a kind of talking – that the set of those who talk quickly is a (probably proper) subset of the talkers" (1982: 161/62). Similarly, if *to Marcia* in (8a) is a function with *speak* as argument, the resulting expression would pick out the set of people speaking to Marcia, without relating it explicitly to the set of speakers – the entailment from (8a) to (8b) remains unexpressed (unless explicitly stated, e.g. as meaning postulates).

61 McConnell-Ginet is discussing in particular Thomason & Stalnaker (1973), but her point is more general.

McConnell-Ginet's second point is that an analysis of adverbs as functors needs to employ a semantic representation involving possible worlds, which, she argues, is the wrong tool to employ for adverbs. Her illustrative example (1982: 162) is a case where in a given model the people who are walking are exactly those who are talking. In this situation, McConell-Ginet argues, an extensional semantics for quickly would imply that those who talk quickly are exactly the ones who walk quickly, a conclusion which might be wrong. This problem can be overcome by having the functor taking the intension rather than the extension of the predicate as argument (1982: 162):

(9a) *walk quickly* translates into **(quickly(^walk))**

(9b) *talk quickly* translates into **(quickly(^talk))**

By introducing intensions, *walk* and *talk* are interpreted not with respect to the original model (where their extensions are still identical), but with respect to possible worlds (or situations), where the people who are walking are not identical to the people who are talking (otherwise *walk* and *talk* would indeed mean the same thing). In this way, the results of applying *quickly* to *walk* and *talk* are differentiated. Now, McConnell-Ginet's criticism of this solution is that it does not provide an explanation for what one intuitively thinks of the difference between walking and walking quickly, namely that *quickly* specifies *walk*, indicating a special way of walking, say, 'at a fast rate'. A similar observation holds for the difference between cooking and eating on the one hand, and cooking fish and eating fish on the other. In a similar scenario, where cooks and eaters are the same people, the thing to do in order to get the contribution of *fish* is to think of a further specification of the situations of cooking and eating: "Cooking fish and eating fish can be distinguished in a model that does not distinguish cooking and eating, with no appeal to alternative situations" (1982: 163). The conclusion from this is that "intensions are beside the point here" (1982: 163), but yet alternative situations are the only set-theoretic mechanism to distinguish walking quickly from talking quickly in the situations assumed. The conclusion from that is that adverbs should not be treated as functions on predicates.

The alternative proposal McConnell-Ginet develops models adverbs as arguments of verbs. It does not only cover lexical adverbs, but can be extended to include optional arguments. The relevant basic features are, first, that VP-internal adverbs combine with verbs rather than with a VP, secondly, that adverbs typically have a dual function – they augment the order of the verb with which they combine and they specify the value(s) of the added argument place(s), and thirdly, adverbs combine with verbs independently of verbal subcategorization (McConnell-Ginet 1982: 167)[62]. That is, for example, an optional PP combines with a verb by adding an argument position to the verb and specifying at the same time how it should be filled (e.g. by a 'location' if it is a locative adverb). This process is freely available, independent of the subcategorization of the verb.

The formal statement of this approach makes use of 'multiple-order predicates', a possible translation of verbs like *speak*, which can be used as intransitive and as transitive verbs. The verb *speak* could be translated as 1/2-order predicate with the restriction in (10) (McConnell-Ginet 1982: 168), where 'Den' abbreviates the denotation of a predicate in particular situations:

(10) if $<a, b> \in$ Den(**speak**), then $\{a\} \in$ Den(**speak**)

That is to say, if an ordered pair of constants, meaning, maybe, Anne and Berta, is in the extension of speak', then so is the first member of the pair, that is here, Anne. Under this interpretation, *speak* can be used with or without PP, and the inference from (11a) to (11b) is valid:

(11a) Anne spoke to Berta.
(11b) Anne spoke.

But rather than just translating some verbs as being of multiple order, McConnell-Ginet develops a formulation with which multiple-order verbs can be formed by adverbs, which 'augment' the verb. This is defined formally as (McConnell-Ginet 1982: 169)[63]:

(12) *Admissible Augmentation (McConnell-Ginet 1982: 169)*

> Let α be a verb in category X that translates into an n-order predicate denoting an n-ary relation $\mathbb{R}$. Then α^+ is an Admissible Augmentation of α in categories X and X/Y only if α^+ translates into an $n/n+1$-order predicate denoting $\mathbb{R}^+ = \mathbb{R} \cup \mathbb{S}$, where $\mathbb{S} \leq \mathbb{R} \times$ Type Y. The augmented verb α^+ is admissible relative to $\xi \in$ Y only if $\mathbb{S} \leq \mathbb{R} \times \mathrm{Den}(\xi) \neq \emptyset$.

62 A fourth characterisic is that VP adverbs are basic and that their function as higher-level adverbs can be defined by meaning postulates. This does not carry over to e*.
63 Note that '≤' here and throughout this chapter indicates a subset relation.

The definition in (12) says that, for example, a binary predicate can be formed from a unary predicate if it can be augmented by a modifying expression such as an adverb. The denotation of this new predicate ($\mathbb{R}^+$) is formed from the union of the denotation of the original, unaugmented verb ($\mathbb{R}$) with the denotation obtained from the product of $\mathbb{R}$ and the 'categorial range' (Type Y) of the modifier ($\mathbb{S}$). The categorial range of manner adverbs, for example, is 'manner', that is, the added argument is restricted relative both to the verb's meaning and the 'kind' of modifier. A verb may be augmented by a modifier only if there is at least one member in the union of the denotation of verb and modifier ($\mathbb{S} \neq \emptyset$). Thus for the example (4), above, rather than translating *speak* as multiple-order predicate, it can be translated as a unary predicate, because by (12) it can become a multiple-order predicate if it is used with a modifier (here, the PP). The step from Anne spoke' (speak unary) to Anne spoke+' (speak binary) is licensed (admissible) if there is somebody who Anne spoke to, as for example Berta. In this scenario, the denotation of the new predicate spoke+' is both {Anne}, the denotation of the original verb, and {<Anne, Berta>}, the union of verb and modifier denotations, where the categorial range of the modifier would be 'addressee'.

With this formulation of augmenting a verb, optional adverbial modification can be formulated as follows (McConnell-Ginet 1982: 169)[64]:

(13) *Ad-Verb Rule (McConnell-Ginet 1982: 169)*

> Let α be a lexical verb belonging to category X where X $\neq$ Y/AD-V and ξ be an expression belonging to category AD-V. Then expression $\alpha\xi$ belongs to category X. A translation of $\alpha\xi$ is defined if $\exists\alpha^+$, an admissible augmentation of α relative to ξ in categories X and X/AD-V. Then $tr(\alpha\xi) = tr(\alpha^+)(tr(\xi))$.

With the rule in (13), any verb of any category (i.e. intransitive, transitive, etc.) can be augmented by an expression of category AD-V, that is, if it is an Ad-Verb, a type which includes those expressions which (can) function as optional arguments of a predicate and thus modify it. The combination of the verb and

---

64 There is a corresponding rule for obligatory adverbial modification, for cases like (McConnell-Ginet 1982: 166):

(i) Liz resides in Kalamazoo
(ii) *Liz resides

This is irrelevant here since I treat PPs as Ty(e) (in contrast to Ty(AD-V)), so that (i) does not need a special rule (other than lexical).

the Ad-Verb (e.g. by function application) yields an expression of the same category as the original, unaugmented verb – both α and $\alpha\xi$ belong to category X (hence the rule is recursive). The interpretation of this rule (its translation) restricts its application, since a translation is only defined if there is an admissible augmentation (as defined above). If there is, then the interpretation of the complex expression $\alpha\xi$ is defined as the translation of the augmented verb (as defined above) applied to the translation of the Ad-Verb.

To consider an example, if *speak* is an intransitive verb of type Ty(e –> t) (i.e. <e,t> in Montagovian notation) denoting speak' and *to Linda* is an Ad-Verb of type Ty(AD-V) of 'addressee' Ad-Verbs, then an expression *speak to Linda* of type Ty(e –> t) can be created. The interpretation of this new expression is the interpretation of *speak*, i.e. in terms of the lambda calculus Fo(λx speak'(x)), plus the augmentation, so that α^+ here would be Fo(λx speak'(x) & λyλx speak'-addressee'(y)(x)), which combines with (*to*) *Linda* such that Linda' fills the second variable slot:

(14) $\underline{(\lambda x\ \text{speak}'(x)\ \&\ \lambda y \lambda x\ \text{speak}'\text{-addressee}'(y)(x))\ (\text{linda}')}$

$(\lambda x\ \text{speak}'(x)\ \&\ \lambda x\ \text{speak}'\text{-addressee}'(\text{linda}')(x))$

That is, under this view, Ad-Verbs create an additional argument slot of the predicate which they subsequently fill. In this sense, the Ad-Verb and Augmentation rules describe a two-step process, creating and filling of argument slots. Since the meaning of the augmented verb is built on, and incorporates the unaugmented verb, the inference from *Mary spoke to Linda* to *Mary spoke* is explicitly recorded in the semantics. By assuming different kinds of Ad-Verbs, the Augmentation rule ensures that only semantically 'suitable' Ad-Verbs can augment predicates.

McConnell-Ginet concludes that this characterization of Ad-Verbs overcomes the problems created by treating adverbials as functors. It takes account of the fact that Ad-Verbs modify their base verb, and that the meaning of the base verb is entailed by the augmented verb. The semantics is both compositional and extensional, this latter quality being argued to be more suitable than an intensional treatment. Finally, the proposal draws on certain similarities between lexical adverbs and optional term phrases (NPs and PPs), and provides thus an analysis for both cases.

The analysis presented here is similar to McConnell-Ginet's proposal with respect to the role of optional Ty(e) expressions in the verb phrase, so that I will use the Augmentation and Ad-Verb rules when I propose an extensional semantics for e^* in Section 3. Before presenting this discussion, however, the next subsection presents a digression from the main argument developed here – I discuss a third way of developing a semantics for adjuncts, which in a way overcomes the dichotomy between the functor and the argument approach, namely Minimal Recursion Semantics. I show how this semantics can be used for e*, but argue that this does not present a conceptually interesting alternative, so that I use McConnell-Ginet's approach as a starting point for a semantics for e*.

2.3. Adjuncts as Scope

The idea advocated in the preceding chapters, that verbal subcategorization is underspecified, has been explored in models other than LDSNL. In fact, part of the discussion in Chapter 2 was based on work conducted within Head Driven Phrase Structure (HPSG)[65]. In this section, I trace the formalization of this observation in HPSG, and present the corresponding semantics, Minimal Recursion Semantics (MRS). As will be seen, MRS provides an alternative to the two analyses of the semantics of adjuncts introduced so far. Although eventually I do not adopt the ideas developed in MRS, I discuss the system here because it presents a pertinent proposal in the (formal semantics) literature. The section will serve furthermore as a more general comparison of LDSNL and HPSG with particular reference to the topic discussed here. The section proceeds as follows: I offer a brief introductory discussion of the relation between LDSNL and HPSG (2.3.1.) and then turn to the HPSG/MRS analysis of verbal underspecification (2.3.2.). I develop a semantics of e* based on MRS (2.3.3.) and discuss why I do not develop this line further (2.3.4.).

2.3.1. LDSNL and HPSG

The main distinctive quality of LDSNL with respect to alternative conceptualizations in linguistics is the assumption that natural language grammar reflects comparatively directly the fact that grammatical knowledge is used by hearers in the incremental building of interpretations for strings of words. The picture drawn in LDSNL is thus performance (in particular, hearer) orientated and representational, so that ultimately a psychological claim is being made. In contrast, HPSG, which has its roots in the formal semantics tradition, is less concerned with cognitive claims, but rather develops a formal

65 In particular Bouma, Malouf & Sag (1997) and Sag (1998). See Pollard & Sag (1994) and also Horrocks (1987) and Lappin & Johnson (1999) for discussions of HPSG/GPSG and its relation to other theoretical approaches.

apparatus to capture the relation between syntactic strings and their semantic form directly, as well as grammatical well-formedness conditions. While influenced in particular by computational approaches to linguistics, the basic HPSG architecture is neutral between parsing and generation, so that there is no particular reflex of parsing in the formulation of the basic grammar.

However, with respect to formalization, LDSNL and HPSG are less distinct than one might expect from their different theoretical approaches. Basic underlying HPSG representations (here a non-technical term) are typed feature structures (Carpenter 1992), where syntactic and semantic information of words and morphemes is expressed by features which have particular values, and combination of words into phrases are modelled as feature interaction, in particular feature unification. Within this general approach, functor–argument relations are captured as heads and daughters: a head inherits features from its daughter(s), a notion which corresponds to the syntactic–semantic types of Categorial Grammar. The versatility of HPSG results from the fact that there is no theoretical restriction of what features are postulated, nor, in principle, a restriction on feature interaction. Given the head–daughter relation, HPSG feature structures can be represented as binary trees, and any (ordinary) binary tree can be represented as a typed feature structure.

The relation between trees and structural desription in LDSNL is rather similar to HPSG, since any LDSNL tree can be represented as a tree description employing the basic vocabulary for tree description introduced in Chapter 1. So for example, a simple tree like (15) can be represented as (16):

(15) *Tree for* "John saw Sally"

```
                          • {Tn(0), Fo((see'(sally'))(john')), Ty(t)}
                         / \
              • {Tn(00),  • {Tn(01), Fo(see'(sally'))}
     Fo(john'), Ty(e)}              / \
                          • {Tn(010),    • {Tn(011), Fo(sally'), Ty(e)}
                            Fo(see')}
```

(16) *Tree Description for* "John saw Sally"

{Tn(0), Fo((see'(sally'))(john')), Ty(t), Ts($s_i < s_{utt}$), $<d_0>$ (Fo(john'), Ty(e)), $<d_1>$ (Fo(see'(sally')), Ty(e –> t)), $<d_1><d_0>$ (Fo(see'), Ty(e –> (e –> t))), $<d_1><d_1>$ (Fo(sally'), Ty(e))}

The tree in (15) can thus be converted into the single albeit long statement in (16) by exploiting the modality operator <d>, which means that the tree in (15) is described in (16) as seen from the perspective of the top node. In general, an exhaustive description of any LDSNL tree can be given as a set of annotated nodes with annotations. I have also included, for reasons which will be clear shortly, a predicate Ts (tense) in (16) which states that the time of the event described by the formula value precedes the time of utterance.

To bring out the analogy to HPSG more clearly, one might consider the LDSNL Predicates (like Tn, Fo, Ty, etc.) as features. For example, the predicate Fo corresponds rather closely to the HPSG feature CONT, where 'lexical' semantic information from words is recorded, and values of daughters (i.e. LDSNL arguments) are recorded in the CONT feature of the head (i.e. LDSNL functors). In this case, function–application in LDSNL is rendered as feature inheritance. For other predicates, there seems to be even less difference – this was the reason for including the Ts predicate in (16). The value of the tense predicate is recorded at the root node by pointer movement; from the point in the tree where temporal information is provided, the pointer is licensed to go up to the root node, record a value, and return to the point of origin. The pointer movement in this case is not related to predicate–argument structure, function–application, or any of the computational rules of the LDSNL system. In that sense it resembles closely the feature unification employed in HPSG[66]. However, the analogy is not so close for the LDSNL type predicate, where no simple translation is available, since HPSG employs a number of features to describe combinatorial properties, e.g. SUBCAT(egorization), VAL(ence), SUBJ(ect), COMP(lements). This difference reflects the fact that the notion of subject is defined structurally in LDSNL as the last argument to a functor yielding Ty(t) (or, alternatively, as the argument node dominated directly by Tn(0)), and not directly as predicate value. The more general observation is here that a number of structural notions are not recognized in HPSG. This includes in particular the structural underspecification modelled as <d*>, and the LINK relation between (sub-)trees. However, given the isomorphism between trees and tree description, there is, I believe, no principled reason which would prevent the translation of <d*> and LINK into HPSG, although that would probably result in LDSNL in HPSG terms, rather than in HPSG as we know it, the difference being what one chooses to formalize, rather than what one can formalize. Although more could be said at this point, I conclude

66 I do not discuss the details of tense and temporal information here. The interpretation of tense in LDSNL is discussed in Perrett (1996, fcmg.).

here by stating what I wanted to show with this brief comparison, and what I want to use it for. From what has been indicated here, two points follow, namely that LDSNL and HPSG are similar with respect to large parts of the formalization (except possibly for the underspecification), but that, on the other hand, there is a significant difference in conceptual content[67]. I will concentrate on the formal similarities throughout most of this section, where I develop an underspecifed semantics for e* by 'translating' Minimal Recursion Semantics (MRS), developed with HPSG in mind, into LDSNL, but the conceptual differences are addressed at the end of this section.

2.3.2. Minimal Recursion Semantics

Although structural underspecification is not expressed in HPSG, there are proposals to express argument–adjunct symmetries, similar to the proposal made here. The syntactic statement to achieve this is in fact more straightforward than the e* formalization, partly because incrementality plays no role, and partly because features may be introduced rather liberally as long as warranted by the data. Following Bouma, Malouf & Sag (1997), Sag (1998) proposes a lexical rule by which a verb can take any modfier occuring with it as an argument[68] (Sag 1998: 10):

(17)

$$\begin{bmatrix} \textit{vb-lxm} \\ \text{ARG-ST } \mathbf{1} \end{bmatrix} \Rightarrow_{LR} \begin{bmatrix} \textit{word} \\ \text{HEAD } \mathbf{3} \\ \\ \text{ARG-ST } \mathbf{1} \oplus \text{list}\begin{bmatrix} \text{MOD} \begin{bmatrix} \text{HEAD } \mathbf{3} \\ \text{CONT } \mathbf{2} \end{bmatrix} \end{bmatrix} \\ \\ \text{CONT } \mathbf{2} \end{bmatrix}$$

The lexical rule in (17) states that for any given verb with a given argument structure, a new word can be derived which is like the input verb, but where the argument structure is supplemented with a list of modifiers. The amend symbol ('⊕') indicates an addition to a list, since argument struture in general

67 I assume here tacitly that theories can be distinguished on more grounds than empirical coverage, cf. Davidson (1999).

68 A number of alternative formulations are discussed in Sag (1998) and elsewhere, some of them restricting the introduction of modifiers as arguments to slash values (i.e. extracted modifiers), some of them running on a separate DEP(endent) feature, rather than on argument structure. The formulation reproduced here comes closest to e*

is modelled as a list[69]. The head feature of the modifier unifies with the head feature of the derived word, that is, whatever the (syntactic) head of the modifier (for example, a preposition) is, it is dominated by the verb, and, similarly, the content of the modifier is represented in the content list of the derived form.

The syntactic effect of (17) is similar to the e* formalization developed in the last chapter. Under both formalizations a sentence like (18) will have three arguments:

(18) The Sheriff of Nottingham sentenced Robin Hood for three years

In both LDSNL and HPSG (with e* and (17) respectively), the modifier *for three years* can be analysed as an argument of the verb *sentence* under the same structural configuration as *Robin Hood* is, at some level of syntactic description. The difference between the two approaches is that (17) is formulated as a lexical rule, while e* is defined syntactically and does not change lexical information. Furthermore, given the comparative formal richness of HPSG, the impact of (17) can be constrained by additional features where syntactic or semantic valency can be stated (e.g CONT, VAL, COMP), while corresponding predicates are not readily available in LDSNL[70].

Irrespective of the differences between the two formalizations, both do, or may, result in a struture where *sentence* appears as ditransitive:

(19) (((sentence'(for_three_years'))(RH'))(sheriff_of_n'))

The representation in (19) is a LDSNL with e* representation, but an HPSG representation of (18) would be identical in the relevant respects, in particular since the semantics which goes with it presupposes even less syntactic structure than is given in (19) – Minimal Recursion Semantics is defined over flat verb phrase structures, so that the functor–argument structure inherent in (19) can be removed:

(20) sentence(sheriff_of_n', RH', for_three_years')

69 In the present context this can simply be seen as addition. Formally, amend is 'weaker' than set union since it not only does not impose order (which has to be specified additionally if desired) but also allows for doublets, i.e. two identical expressions are two elements.
70 With the exception of lexical instructions.

In fact, MRS allows for a flat representation of sentences, represented as simple lists under conjunction (symbolized here as '/ \'):

(21) / \ {sentence', sheriff_of_n', RH', for_three_years'}

MRS is a semantics which makes it possible in principle to represent semantic scope irrespective of the order of elements in the predicate. Before seeing how this works exactly, it is useful to state the objectives MRS is trying to achieve. MRS as described by Copestake, Flickinger & Sag (1997) (CFS) is developed for computational semantics, with the particular aim of providing a clear link between syntax and semantics, to express the relation between the meaning of the whole and the meaning of its parts, and to allow for underspecification (CFS 1997: 1). This latter requirement refers not so much to underspecification as discussed here, but to cases where a fully resolved resolution of, for example, quantifier scope interaction is either not necessary, or not possible[71]. Given the application of MRS in computational linguistics, this might arise in a case where a fully scoped structure would involve "arbitarily complex reasoning on completely open-ended world knowledge" (CFS 1997: 2).

The most extensively discussed examples in CFS are sentences involving quantifier scope, such as the sentence in (22a), where scope can be represented as in (22b):

(22a) Every tall man is old
(22b) every(x, man(x) & tall(x), old(x))

The corresponding flat representation is (23):

(23) / \ {every(x), man(x), old(x), tall(x)}

But the problem with (23) is that it is not only the representation of (22a), but also of (24):

(24) Every old man is tall

71 The notion of underspecification here is closer to underspecified Discourse Representation Structures (UDRS), cf. e.g. Reyle (1993), König & Reyle (1998), an intermediate structure between natural languages strings and DRS's proper. I omit a discussion of UDRS for reasons of space.

That is to say, completely flat structures obliterate scope distinctions encoded in syntactic structure, and, as CFS argue, need to be enriched so as to preserve scope information.

This enrichment of flat structures is achieved in MRS by introducing a feature HANDLE into the feature structure of every expression, which serves to make reference to it in the semantics:

(25) top 0
/ \ {0: every(x, 1, 2), 1:man(x), 1:tall(x), 2:old(x)}

In (25), all words have been assigned a numerical index, corresponding to the value of the HANDLE feature. With this – albeit informal – representation, it is possible to state the scope restrictions expressed by (22a) (and, by changing the indices, those expressed by (24)). The value of the 'top' node indicates that the expression with the index 0, here 'every(x, 1, 2)', is outermost, so that it outscopes all other components of the list (i.e. the representation of the sentence). The representation of *every*, like those of all quantifiers in MRS, is given as a generalized quantifier, with bound variable, restriction and scope[72]. The fact that *tall*, rather than *old* as in (24), is in the restrictor of *every* is captured by the fact that the index of *tall* is 1, while the index of *old* is 2, matching the index in the scope of *every*. Although the actual HPSG/MRS formalization of HANDLE (and two related features, LISZT and KEY) looks rather different from (25)[73], the basic idea is that by providing an extra feature, any expression in a sentence can be related to any other expression as being identical to it in relevant respects, as well as being dependent on it. Furthermore, co-indexation of a HANDLE value with the top value results in wide scope reading. Since all this is formulated as values of the HANDLE feature, underspecification can be modelled by defining variables as values for HANDLE. Thus, in (26b) p, n, m and are variables over handles, and the MRS structure is underspecified with respect to object or subject wide scope. The scoped representations are given in (26c) (wide scope subject) and (26d) (wide scope object):

72 See Barwise & Cooper (1984), Keenan (1996), Cann (1993: 187ff.) for discussion of generalized quantifiers (GQ's).
73 MRS structures are, like HPSG structures, defined as feature structures. However, I do not represent more technical details than are necessary for the present discussion.

(26a) Every dog chased some cat

(26b) top p
/ \ {1:every(x, 3, n), 3:dog(x), 7:cat(y), 5:some(y, 7, m), 4:chase(e, x, y)}

(26c) top 5
/ \ {1:every(x, 3, 4), 3:dog(x), 7:cat(y), 5:some(y, 7, 1), 4:chase(e, x, y)}

(26d) top 1
/ \ {1:every(x, 3, 5), 3:dog(x), 7:cat(y), 5:some(y, 7, 4), 4:chase(e, x, y)}

As can be seen from (26c) and (26d), the two different readings of (26a) are captured by choosing different values at the top node. As can also be seen, the choice of which handle is top implies the choice of the value of the scope of the quantifiers. More generally, there are a number of constraints on possible MRS structures in general, as well as sentence-particular constraints on particular value combinations. These constraints are expressed as relations between a pair of HANDLE values, namely either *equivalence*, i.e. the values are identical, or *outscope*, i.e. one value has scope over the other. In general, any MRS structure is fully specified if the handle values are such that a path can be traced from the top value through all handles and that the resulting structure is a tree. In addition, for any MRS structure, any constraints on handles (i.e equivalence or outscope) can be established in the incremental unification of feature structures (i.e. in the syntactic computation of the sentence).

In a scoped MRS structure, variables are generally bound by quantifying expressions. However, for certain variables, 'implicit toplevel' binding is possible, that is, the values are immediately dependent on the top node. The variables which allow for implicit toplevel binding are event variables, a variable introduced by a proper name or a pronoun, or a variable which occurs in a position where it corresponds to a syntactically optional argument. The former two cases will not be discussed any further, but the last type of variable is important for the present discussion. What implicit toplevel binding effectively means is that the scope analysis of quantifiers is extended to the relation between adjuncts and verbs, so that predicate–argument relations are analysed as scope relations. Recall that functor–argument relations in HPSG are expressed as features and feature unification, similar to the representation of scope dependencies within MRS. On this analogy, the difference between treating an adjunct as a functor or as an argument with respect to the verb can be expressed as simple restrictions on handles, by outscope. That is, the sentence in (27) can be syntactically represented as flat (as in (28)), provided that predicate argument relations can be expressed as handle interaction (CFS 1997: 4)[74]:

(27) On Monday Kim ran in Foothills Park.

(28) / \ {on(e, Monday), run(e, Kim), in(e, Foothills_Park)}

The extension of MRS to adjuncts provides a means for developing a semantic analysis of adjuncts, which is to a certain extent independent of the syntactic arguments discussed with respect to Adjunction and e*. Thus, given that, on the one hand, adjuncts can be analysed in HPSG as (optional) arguments, namely by the lexical rule presented in (17), above, and that, on the other hand, optional arguments permit toplevel binding[75], it seems plausible to test the possibility to develop a semantic analysis for e* on analogy with MRS. This will be done in the next section.

2.3.3. Minimal Recursion Semantics for e*

Given the areas of overlap between HPSG and LDSNL with respect to formal architecture discussed above, it seems viable to adopt some aspects of MRS and exploit them for the development of rules for the semantic evaluation for underspecified verbs. Recall that the reason for discussing the semantics of adjuncts, and for introducing MRS, was the observation put forward in the last chapter that in order to evaluate the advantages and problems of underspecified verbs, an account of the interpretation of structures like (29b), derived with e* from (29a), should be given:

(29a) John was singing with Mary in the kitchen

(29b) {Tn(0), Ty(t), Fo(((sing'(in_the_kitchen'))(with_mary'))(john'))}

MRS appears to offer another possibility to tackle that problem – I will propose in this section that the adjuncts in (29), *with Mary*, and *in the kitchen*, while

---

74 CFS (1997: 3) introduce these examples for expository purposes only, not as serious proposals for analyzing temporal adverbs. Yet they presumably illustrate the scope of the semantics. Examples like this are discussed below in the LDSNL version of MRS. Note that the argument here does not depend on the use of the event variable.

75 The relation between MRS and optional arguments has been discussed further in conference presentation by Copestake (1996), but which has, to my knowledge, not been published.

syntactically behaving like arguments, can semantically be treated as toplevel bound adjuncts. In order to make this work, some modifications of the e* analysis developed so far are introduced and discussed. Note, however, that MRS as it stands is not exclusively designed for HPSG – it should be possible, in principle, to model all semantic interpretation in LDSNL as being defined by MRS, although it is not clear what the benefits of this approach would be. In the following discussion, I am thus only concerned with providing a means to amend the e* analysis of the last chapter so as to make it compatible with MRS, which then provides a semantic interpretation in which the relation between verbs and arguments is analysed as a scope relation. After introducing the necessary modifications, the results of such an analysis are discussed and evaluated.

To start with, two modifications are required – the introduction of 'handle' values, and a top level predicate where adjuncts can be projected.

Since in LDSNL, all tree positions are uniquely identified by their tree address, one could use these addresses as handles. However, I introduce a new predicate named Handle (Hd(x)) to make the modification more explicit, and I furthermore assume that its value is part of the semantic content of declarative units, that is, it is part of the formula value. I include handle predicates for expressions at terminal nodes, in the order in which they are introduced into the tree, and provide a predicate holding at the root node, where all handle values introduced into the tree are recorded. Thus, for example, (30) is a declarative unit with handles:

(30) {Hd(1), Fo(1: sing), (Hd(2), Fo(2: john)}

In order to model top binding of adjuncts, i.e. Ty(e) expressions introduced by the resolution of underspecification, I assume that handle values are projected directly at the top node, Tn(0). This can be formally expressed by introducing a predicate $\sigma(\{x, y, z, \ldots\})$, where handle values are recorded. Finally, handle values are recorded at the root node by lexical actions. The rule in (31) defines that lexical action addHandle:

(31) *Handle Rule*

(1) IF T
THEN ... add_handle ...

where T an arbitrary trigger

(2) AddHandle: 1) put(Hd(x)), where x = 1, 2, ... and x is fresh
2) $\sigma := \sigma \cup (x)$,
where σ is the set of handles at Tn(0).

The first clause of the Handle rule states that add_handle is a possible lexical action. The second clause of the rule states that the action to be carried out consists in introducing a handle predicate and a new value for it, i.e. a value which has not yet been assigned in the tree, and, secondly, to introduce the value into the set σ where all handle values which have been assigned are recorded[76]. The rule results in representations where the presence of adjuncts is recorded at the top node Tn(0). This does not interfere with the syntax of e*, since the Handle rule simply adds a new action predicate to the set of actions stated in the lexicon.

A derivation for (29a), repeated here, thus results in a tree like (32) before completion:

(29a) John was singing with Mary in the kitchen

(32) *Tree for* "John was singing with Mary in the kitchen" *(with e* and MRS)*

```
                    • {σ({1, 2, 3, 4}), ?Ty(t)}
                                 /\
• {Hd(1), Fo(1: john'), Ty(e)}      • { ?Ty(e -> t)}
                                          /\
• {Hd(3), Fo(3: with_mary'), Ty(e)}     • {?Ty(e -> (e -> t))}
                                              /\
        • {Hd(4), Fo(4: in_kitchen'),      • {Hd(2), Fo(2: sing'),
                       Ty(e)}              Ty(e -> (e -> (e -> t)))}
```

According to the e* rule, all PPs have been introduced as arguments. The underspecification in the verb has been resolved, and a place in the tree has been assigned. The new feature in the tree is that all expressions are

76 I have not included an explicit statement for pointer movement, which could be added by employing the go predicate and an appropriate modal operator. I also assume that new (fresh) values can be identified.

represented at Tn(0) by their handle value. Since the handle predicate now provides explicit access to Ty(e) expressions, verbal underspecification could under this analysis, i.e. with handles, be modelled as e^n as discussed in the preceding chapter, that is, as flat verb phrase. I retain the e^* formulation here, but note that the formalization with e^n would show the power of the handle predicate even more clearly, and would be closer to the original MRS conception. But even with e^*, it is clear that in a tree like (32) explicit reference can be made to the members of σ, so that semantic scope can be stated independently of the syntactic underspecification.

Of course, as it stands, it is not clear what sort of semantic interpretation is assigned to e^* with MRS – all that has been done is to introduce a new predicate and a lexical rule. But with this modification it is possible to model an analysis of adjuncts both as functors and as arguments if these relations are regarded as scope relations. For example, in (32), *in the kitchen* might be taken as a functor to the predicate, while *with Mary* may be an argument, which can be expressed as handle scoping at Tn(0):

(33) *Tree for* "John was singing with Mary in the kitchen"
(with e/MRS, scoped)*

```
                    • {σ({1, 2, 3, 4}, {4 > 2}, {2 > 3}), ?Ty(t)}
                                        /\
   • {Hd(1), Fo(1: john'), Ty(e)}    • { ?Ty(e -> t)}
                                           /\
 • {Hd(3), Fo(3: with_mary'), Ty(e)}    • {?Ty(e -> (e -> t))}
                                           /\
        • {Hd(4), Fo(4: in_kitchen'),     • {Hd(2), Fo(2: sing'),
                 Ty(e)}                    Ty(e -> (e -> (e -> t)))}
```

The additional constraints over handles express the different semantic functions of the two PPs as scope constraints, where the relation '>' means outscope.

By further developing the account of semantic interpretation for underspecified verbs with handles, a number of extensions could be developed[77]. However, the analysis outlined so far illustrates two points. First, it is not difficult to define the necessary predicates which make it possible to semantically interpret underspecified verbs, and LDSNL trees in general, with

77 For example the interaction of adjuncts and tense, as well as a precise statement of the handle constraints imposed by adjuncts incrementally. However, as outlined in the following discussion, I do not develop this line of inquiry further.

recourse to MRS. On the other hand the formalization is very powerful since it is possible in principle, though probably not in a fully formalized analysis, to obliterate syntactic structure established during the parse, and to replace structures established by function–application by scope relations stated over the handle values introduced by terminal nodes. These two points are briefly discussed in the following section.

2.3.4. Discussion

The preceding discussion shows that a rule of semantic interpretation for e* can be developed by adopting proposals from MRS. The advantage of this approach is that it provides a means to express both an analysis of optional Ty(e) expressions as arguments to the verb, and an analysis of optional Ty(e) expressions as functors taking the verb as argument, namely by employing the outscope relation. However, the disadvantage is that the notions of quantifier, or operator scope and function–application are conflated, and that the dissociation of semantic and syntactic structure implies that the relation between structure established during the parse and semantic combination is obliterated, which contradicts the basic LDSNL assumption that hearers build semantic trees incrementally. Thus, while it is possible to augment the analysis of verbal underspecifcation as e* with a handle feature, the resulting interpretation does not express the structural properties associated with e*, namely that adjuncts can be treated as arguments to verbs because verbal subcategorization is underspecified. The introduction of handles to connect e* verbs with MRS is thus a possible, but not an optimal interpretation for e*.

However, it is worth pointing out that the formalization proposed here does not unduly overstretch the formal apparatus provided by LDSNL. While the introduction of the handle predicate, together with the associated Handle rule, adds to the predicates already recognized, it keeps clear of any unwanted interaction, since it is only used with the equally new σ predicate. Furthermore, there is the possibility, although I have not explored it here, that the handle predicate might in fact be 'reduced' to the tree node identifier. Similarly, the σ predicate is only one in a number of predicates at the root node, such as clausal typing (the 'Q' feature), and, more importantly tense. The handle values are, similar to tense, projected directly at the root node, which implies vacuous pointer movement (in the sense that it is not driven by function–application), similar to feature unification. Since this possibility is independently required, no undue burden is put on the overall system by the introduction of the handle predicates. Finally, the constraints on handles can be introduced incrementally, that is at every step of introducing a new Ty(e) expression, with the verb always having widest scope. In that way, the syntactic mode of combination is mirrored in the scope relation representing semantic interpretation. However, it is less clear if the point about incrementality retains its force if the semantic interpretation is, even if only in principle, defined as being independent of structure building processes. Thus I conclude that the e*/MRS formulation provided here is not problematic with respect to the formal architecture employed in LDSNL, although it is problematic with respect to basic theoretical assumptions.

As noted at the outset of this section, the difference between LDSNL and HPSG is really a matter of theoretical assumptions about (the cognitive basis of) natural language, and not so much one of formal expression. From this perspective, it is not surprising that e* could be combined with MRS. However, e*/MRS has very little to say about why there should be underspecification in the verb phrase, as it makes no reference to the process of building interpretations, or to the role of the hearer. In other words, e*/MRS is not a very 'natural' solution, given the over all assumptions of the system. I will thus not pursue this solution any further.

2.4. Semantics of Adjuncts – Conclusion

I have presented three possible ways of analyzing adjuncts which have been proposed in the formal semantics literature. The first proposal, to treat adjuncts as functors taking predicates as arguments corresponds to the Adjunction rule proposed in Chapter 3, which I have rejected mainly with reference to basic LDSNL assumptions about the incremental nature of tree building. The second proposal, to treat adjuncts as arguments of predicates, corresponds closer to the idea I want to express with e*. It additionally provides further semantic reasons for rejecting the Adjunction rule. Finally, I have presented a third proposal, MRS, which is neutral between Adjunction and e*. I have shown how, on the model of MRS, a semantics for e* can be given, but rejected this formulation on conceptual grounds. The discussion of MRS has served further as an opportunity to compare LDSNL with HPSG. In the following section, I develop semantic rules for e*, based on the analysis provided by McConnell-Ginet (1982), and discuss the question of semantic evaluation and the interpretation of underspecified verbs more generally.

3. Adjuncts as Arguments: Extensional Semantics for e*

In this section I develop an extensional semantics for e* based on McConnell-Ginet's analysis presented in the last section. As has been discussed at various points in the preceding chapters, LDSNL assumes that model-theoretic semantic interpretation is assigned to the proposition expressed, and not to natural language expressions directly. In contrast, all proposals discussed in this chapter assume that the relation between natural language and model theoretic interpretation is unmediated. Since in this section I develop an analysis of semantic interpretation for e* which is based on McConell-Ginet (1982), I suspend the LDSNL assumption about mental representations for the purposes of this section and assume with McConell-Ginet that semantic operations are stated in tandem with syntactic operations. After I have shown how an analysis of extensional semantic interpretation for underspecified verbs can be developed, I discuss the problem of semantic interpretation from the perspective of LDSNL in Section 4.

Out of the three analyses of the semantic interpretation of adjuncts, the proposal by McConnell-Ginet (1982) seems to be the most suitable formulation for developing a semantic analysis for underspecified verbs, since it reflects syntactic predicate–argument structure (in contrast to the more liberal formulation of MRS, where syntactic and semantic structure can be dissociated) and assumes that adjuncts can be analyzed as optional arguments (in contrast to Dowty's formalization which treats them as functors). After introducing the necessary adjustments, I show how the semantic rule for e* works and give a sample presentation. Some problems and evaluations are presented in the last subsection.

3.1. Adjustments and Formulation of a Semantic Rule for e*

As has been noted, the adverb rule(s) proposed by McConnell-Ginet are close in spirit to e*, since adjuncts are treated as arguments to verbs. Semantic interpretation is furthermore established incrementally, in tandem with the introduction of the adverb. In order to retain the basics of McConell-Ginet's proposal, I use the incremental rule e* Partial Resolution in this section and show how it can be used to state a semantic rule for underspecified verbs. The rule e* Partial Resolution is repeated below as (34):

(34) *e* Partial Resolution*

$$\frac{\{_{n^*} \ldots Ty(e^* \rightarrow X)\}, \{_{n0} \ldots Ty(e) \Diamond\}}{\{_{n1^*} \ldots Ty(e^* \rightarrow (e \rightarrow X))\}, \{_{n0} \ldots Ty(e) \Diamond\}}$$

The Ad-Verb rule by McConell-Ginet (1982: 169) is repeated here as (35):

(35) *Ad-Verb Rule (McConnell-Ginet 1982: 169)*

> Let α be a lexical verb belonging to category X where X $\neq$ Y/AD-V and ξ be an expression belonging to category AD-V. Then expression $\alpha\xi$ belongs to category X. A translation of $\alpha\xi$ is defined if $\exists\alpha^+$, an admissible augmentation of α relative to ξ in categories X and X/AD-V. Then $tr(\alpha\xi) = tr(\alpha^+)(tr(\xi))$.

The first difference between the two rules is that the e* rule in (34) is defined for the combination of an expression of type Ty(e* –> X) with an expression of type Ty(e). In contrast, the rule in (35) combines a lexical verb belonging to any category with an expression belonging to the category AD-V. The difference reflects the fact that e* models adjunction as structural underspecification of verbs, and consequently adjuncts as simple nominal expressions of Ty(e), while the rule in (35) does not express any underspecification. Rather, optionality is introduced in part by the recursive nature of the rule, where the expressions to which the rule applies are of 'standard' lexical types, and in part by assigning adjuncts to the lexical type AD-V. Thus, the categories in (35) have to be replaced by LDSNL types, expressing that e* combines with expressions of type Ty(e)[78]. Secondly, McConnell-Ginet's rule assumes the complete parallel operation of syntactic and semantic rules, as implemented in Categorial Grammar ('bottom-top') derivations. Thus the two steps of adverbial modification, the creation and filling of an argument slot are both expressed in the rule. In contrast, e* models the incremental growth of a semantic tree, where the unfolding of tree structure is achieved by the structure building actions of lexical items and the rule prediction, while the actual combination of functors and arguments by Elimination and Completion only takes place when the tree is complete. Thus, the two steps in (35) need to be 'factored out' if the rule is to interact with the syntax of e*. A formulation incorporating these two changes is given in (36):

78 The translation of Ad-V as Ty(e) is further discussed in Section 3.3.1., below.

(36) *Semantic Interpretation of e**

> Let α be a lexical verb of Ty(e* –> X) and ξ be an expression of Ty(e). Then α can be partially resolved to Ty(e* –> (e –> X)). A translation of $\alpha\xi$ is defined if $\exists\alpha^+$, an admissible augmentation of α relative to ξ with types Ty(X) and Ty(e –> X). Then $tr(\alpha\xi) = tr(\alpha^+)(tr(\xi))$.

In (36), the typing reflects the e* approach. The semantic interpretation, on the other hand, is as given in McConnell-Ginet, in that the translation depends on there being an admissible augmentation of the verb. This can be more conveniently expressed by incorporating the semantic requirement into the transition rule for e*:

(37) *e* Partial Resolution with Semantics*

$$\frac{\{_{n^*} \ldots Fo(\alpha), Ty(e^* \text{–>} X)\}, \{_{n0} \ldots Fo(\beta), Ty(e) \Diamond\}}{\{_{n1^*} \ldots Fo(\alpha^+), Ty(e^* \text{–>} (e \text{–>} X)\}, \{_{n0} \ldots Fo(\beta), Ty(e) \Diamond\}}$$

> where $Fo(\alpha^+)$ is an admissible augmentation of $Fo(\alpha)$ relative to $Fo(\beta)$

Note that no reference needs to be made to the interpretation (translation) of the augmented verb since it is covered by the general semantic rule of function–application which is in LDSNL defined by the transition rule Elimination. This means that once the verb is augmented, its semantic interpretation is exactly like the interpretation of a normal predicate with respect to its argument. Since (37) makes reference to admissible augmentations, this rule has to be part of the incremental semantics for e* developed here. It is in its original formulation repeated here (McConnell-Ginet 1982: 169):

(38) *Admissible Augmentation (McConnell-Ginet 1982: 169)*

> Let α be a verb in category X that translates into an n-order predicate denoting an n-ary relation $\mathbb{R}$. Then α^+ is an admissible augmentation of α in categories X and X/Y only if α^+ translates into an $n/n+1$-order predicate denoting $\mathbb{R}^+ = \mathbb{R} \cup \mathbb{S}$, where $\mathbb{S} \leq \mathbb{R} \times$ Type Y. The augmented verb α^+ is admissible relative to $\xi \in$ Y only if $\mathbb{S} \leq \mathbb{R} \times Den(\xi) \neq \emptyset$.

Again, the categories have to be replaced by types:

(39) *Admissible Augmentation for e* (first version)*

Let α be a verb of Ty(e* –> X) that translates into an *n*-order predicate denoting an *n*-ary relation $\mathbb{R}$. Then α^+ is an admissible augmentation of α with Ty(X) AND TY(e –> X) only if α^+ translates into an *n*/*n*+*1*-order predicate denoting $\mathbb{R}^+ = \mathbb{R} \cup \mathbb{S}$, where $\mathbb{S} \leq \mathbb{R} \times$ Ty(e). The augmented verb α^+ is admissible relative to $\xi \in$ Ty(e) only if $\mathbb{S} \leq \mathbb{R} \times \mathrm{Den}(\xi) \neq \varnothing$.

With this modification, the rule interacts with the transition rule for e*, so that the underspecification in e* is partially resolved by expressions of Ty(e) if the verb can be augmented with respect to that expression. With each step of application of the transition rule, i.e. with every new expression of Ty(e), the predicate is extended with an additional argument slot, which is filled with the expression of Ty(e) standardly under Completion. I give a sample derivation in the next subsection and, following that, discuss the implications of the adjustments made.

3.2. Sample Derivation

An example makes clear how the semantic rules work. Consider (40) and the derivation in (41):

(40) Sue was singing with Mary in the shower.

(41a) *Tree for* "Sue was singing

```
                         • {Tn(0), ?Ty(t)}
                                / \
    • {Tn(00), Ty(e), Fo(sue')}     • {Tn(01), ?Ty(e -> t) ◊}

    {Tn(01*), Ty(e* -> (e -> t)), Fo(sing')}
```

At this stage, Sue has been assigned its proper place in the tree, and a node with the requirement TODO Ty(e –> t) has been built. The DU of *sing* is still unattached, pending further input. The lexical actions from the entry for *with* result in a new argument node which is annotated with a prepositional feature and a requirement TODO Ty(e), which is fulfilled when Mary is introduced:

(41b) *Tree for* "Sue was singing with Mary

• {Tn(0), ?Ty(t)}
/ \
• {Tn(00), Ty(e), Fo(sue')} • {Tn(01), ?Ty(e -> t)}
/\
• {Tn(010), Fo(with_mary'), Ty(e) ◊} • {Tn(011), ?Ty(e -> (e -> t))}

{Tn(01*), Ty(e* -> (e -> t))), Fo(sing')}

The tree in (41b) provides a suitable input to e* Partial Resolution, which now not only updates the underspecification of the location and the type value, but also records the semantic operation of augmenting the predicate. An instantiation of the rule in this situation is given below:

(42) *e* Partial Resolution (instantiated as* sing with Mary*)*

$$\frac{\{_{01^*} \ldots \mathrm{Fo(sing')}, \mathrm{Ty}(e^* \rightarrow (e \rightarrow t))\}, \{_{010} \ldots \mathrm{Fo(with\_mary')}, \mathrm{Ty}(e)\ \Diamond\}}{\{_{01^*} \ldots \mathrm{Fo(sing'^{+})}, \mathrm{Ty}(e^* \rightarrow (e \rightarrow (e \rightarrow t)))\}, \quad \{_{010} \ldots \mathrm{Fo(with\_mary')}, \mathrm{Ty}(e)\ \Diamond\}}$$

where Fo(sing'$^{+}$) is an admissible augmentation of Fo(sing') relative to Fo(with_mary')

According to the rule, the predicate is updated with respect to location, type value, and formula value. The operation on the formula value follows from the semantic rule for e*, and can be expressed by lambda expressions. The step from Fo(sing') to Fo(sing'$^{+}$) thus corresponds to the step from (43a) to (43b):

(43a) {Tn(01*), Ty(e* -> (e -> t)), Fo(λx sing(x))}

(43b) {Tn(011*), Ty(e* -> (e -> (e -> t))), Fo(λy λx (sing(x) & sing^{+}(y)(x)))}

As can be seen from (43), the e* semantic rule creates a new lambda term by introducing a new argument slot. Both the original lambda expression and the derived lambda expression represent the meaning of the new verb. This follows from the definition of augmentation, which states that the meaning of the new verb is the union of the extensions of the augmented and the unaugmented verb. The final requirement is that the expression of Ty(e) which triggers the rule denotes something which can fill the new argument slot. From a model theoretic point of view, this means here that the model should contain a predicate Sing^{+}(x)(y). It is in this sense that the semantics is extensional (only one model needs to be considered), which implies also that the semantics is completely static – the semantic expression is evaluated directly against a given model. The implications of this interpretation, and the meaning of admissible, as used in admissible augmentation, are discussed below, so for the moment I assume that in this context, the augmentation is indeed admissible.

The following tree can thus be derived:

(41c) *Tree for* "Sue was singing with Mary

```
                         • {Tn(0), ?Ty(t)}
                                 / \
        • {Tn(00), Ty(e), Fo(sue')}      • {Tn(01), ?Ty(e -> t)}
                                                   /\
  • {Tn(010), Fo(with_mary'), Ty(e) ◊}   • {Tn(011), ?Ty(e -> (e -> t))}


        {Tn(011*), Ty(e* -> (e -> (e -> t))), Fo(sing'+)}
```

The introduction of the second adjunct, *in the shower*, develops the tree in a similar fashion, so that after the relevant actions the adjunct is integrated at a functor node. This situation then provides the input to another application of e* Partial Resolution to result in the following tree:

(41d) *Tree for* "Sue was singing with Mary in the shower

```
                         • {Tn(0), ?Ty(t)}
                                 / \
        • {Tn(00), Ty(e), Fo(sue')}      • {Tn(01), ?Ty(e -> t)}
                                                   /\
  • {Tn(010), Fo(with_mary'), Ty(e)}    • {Tn(011), ?Ty(e -> (e -> t))}
                                                           /\
          • {Tn(0110), Fo(in_the_shower')),      • {Tn(0111),
                                 Ty(e) ◊}          ?Ty(e -> (e -> (e -> t)))}

        {Tn(0111*), Ty(e* -> (e -> (e -> (e -> t)))),
        Fo(λz λy λx (sing(x) & sing+(y)(x) & sing++(z)(y)(x))}
```

This step is formally identical to the preceding one. The predicate has now been augmented twice so that the formula value includes three different predicates of three different arities. In the absense of further lexical input, Merge applies and the unfixed node can be incorporated into the tree. The application of Completion and Elimination then yields the following lambda expressions:

(41e) *Tree for* "Sue was singing with Mary in the shower"

• {Tn(0), Ty(t), Fo(sing(sue') & sing^{+}(with_mary')(sue') &
sing^{++}(in_the_shower')(with_mary')(sue')}

/ \

• {Tn(00), Ty(e), Fo(sue')} • {Tn(01),
Ty(e -> t),
Fo(λx (sing(x) & sing^{+}(with_mary)(x) &
sing^{++}(in_the_shower')(with_mary)(x))}

/\

• {Tn(010), Fo(with_mary'), Ty(e)} • {Tn(011),
Ty(e -> (e -> t)),
Fo(λy λx (sing(x) & sing^{+}(y)(x) &
sing^{++}(in_the_shower')(y)(x)))}

/\

• {Tn(0110), Fo(in_the_shower')), Ty(e)} • {Tn(0111),
Ty(e -> (e -> (e -> t))),
Fo(λz λy λx (sing(x) &
sing^{+}(y)(x) & sing^{++}(z)(y)(x))}

This sample derivation shows how the rule for e* with semantics works, and that it results in an array of predicates with different arities, following the definition of augmentation. This treatment is incremental, in that the predicate is augmented at every introduction of a Ty(e) expression, compositional, in that syntactic and semantic rules are defined in tandem, and extensional, in that the resulting expressions can be evaluated in a first order model by relating the ordered tuples denoted by the augmented verbs to the extensions of the relevant predicates. In the following section, I discuss some implications of the semantics for underspecified verbs developed here.

3.3. Discussion

As noted at the beginning of this section, I have assumed here that semantic interpretation can be assigned to natural language expressions directly, so that I could build on McConell-Ginet's work to develop a semantic rule for underspecified verbs. In this section, I discuss the rule developed in this section in the light of the LDSNL assumption that semantic interpretation is assigned to the propositional form derived, but not to the string directly. From this perspective, the notions of categorial range, entailment, and admissibility employed in the semantic rules for e* receive a different interpretation. However, it will also be shown that, rather than defining the semantics of e* in model-theoretic terms, the interpretation of underspecified verbs can be characterized more correctly as their contribution to the establishment of the proposition expressed, which is the topic of the following chapter.

3.3.1. Categorial Range

In the translation of McConnell-Ginet's Ad-Verb Rule into a semantic rule for e*, I have translated the original semantic type Ad-V as Ty(e), since I have assumed that PPs augmenting predicates are of Ty(e). The original type Ad-V does include PPs, in addition to adverbs, although McConnell-Ginet does not fully develop her original proposal to include PPs functioning as adjuncts. The type Ad-V fulfills a similar function to what I have analysed as the function of prepositions, in that it licenses an expression to be introduced into the verb phrase. However, I am here less concerned with the syntactic aspects of licensing, but rather with the semantic aspects of the analysis of verb modification discussed here. In particular, I am concerned with a possible objection to my translation of the Ad-V rule, namely that the semantic contribution of expressions of type Ad-V and Ty(e) expressions might differ, even though both expressions act as argument of verbs in the relevant cases discussed here. The semantic rule would thus not just increase the arity of the predicate, but also ensure that the augmentation is justified with respect to the categorial range of the expression with which the predicate is augmented, so that the translation of Ad-V to Ty(e) does not express adequately the contribution of the modifier. However, this objection is not justified, since the notion of categorial range in the case of verb modification is not precise, as I try to show in this section. The lack of precision is in fact part of a more general problem related to the semantic contribution of optional modifiers, which is also found in the analysis of Dowty (1979) discussed above.

The relevant point is that both McConnell-Ginet and Dowty assume that PPs, be it as functors or as arguments, can be analysed as specifying lexically in which 'thematic' relation they stand to the expression they modify. The term thematic in this context is mine, and both authors handle this problem differently, but both imply that PPs and adverbs come with some information as to whether they relate to time, location, manner etc. For example, temporal adverbs for Dowty are members of a special category (TmAV), although no categories are proposed for other ways of modifying[79]. McConnell-Ginet assumes that expressions in the category Ad-V provide information about whether they refer to manner, or to 'addressee', etc. Although this is not made precise, it is assumed in the 'two-step' analysis of adverbial modification, where the additional argument slot is 'annotated' with an appropriate label, and a given verb–ad-verb expression is admissible if the modifier denotes a member of this particular type of modification. The absence of a more formal characterization of this information indicates to me the more general problem of incorporating into formal semantics what in syntax is often referred to as thematic roles. As already discussed in Chapter 2, I do not assume thematic roles here. Semantically, it seems doubtful to me whether it is possible to define the thematic contribution of adjuncts lexically, as seems to be tacitly assumed by Dowty and McConnell-Ginet. Rather, thematic information results from the meaning of the predicate, the preposition, and the object of the preposition, so that it seems plausible to view thematic information as something which can be inferred from the proposition expressed, rather than something which is lexically or syntactically 'primitive' (cf. Ladusaw & Dowty 1988). From this perspective, thematic information has no special status, but is part of the general phenomenon of inferential activity in utterance interpretation.

The argument presented in this section is thus that the type Ad-V cannot be lexically defined if it is to imply the contribution of thematic meaning, from which it follows that the translation of Ad-V as Ty(e) in the semantic rule for e*is justified. This in turn entails that augmentation of predicates is the semantic process by which optional Ty(e) expressions are introduced into the augmented verb as arguments, corresponding to the formalization of verbal underspecification with e*.

3.3.2. Entailments

The second point to be discussed with respect to the semantic rule for e* is the proliferation of predicates which results from the application of the rule. I have up to now assumed that McConnell-Ginet's motivation to encode in the semantics the fact that if John talks to Mary, he talks, is correct. This is formally expressed by having the extension of an augmented verb to be the set union of

79 The prominent place of temporal modification in Dowty (1979) I believe partly reflects the fact that tense, together with modality, are the best worked-out cases of possible world semantics, whereas there is no correspondingly developed possible world semantics of places, instruments, or 'accompaniment'.

both the unaugmented and the augmented verb, together with the corresponding addition of lambda terms. However, as briefly hinted at in the discussion of Dowty's semantics, it is not clear how to model the combined effect of the lexical meaning of the verb, semantics, and pragmatics in these cases. From the perspective adopted in LDSNL, entailments like those encoded in the definition of admissible augmentation are not encoded in the semantics, but rather are analysed as inferences which can be derived from the eventual semantic representation, the propositional form. Given the overall Relevance theoretic perspective of utterance interpretation, the semantic rule as it stands would imply that all propositions which result from the application of the rule are in fact communicated and should contribute to the derivation of relevant inferential effects. Since this is very implausible, the inclusion of entailment relations into the semantic rule for e^* is inappropriate.

Furthemore, even if it this theoretical objection is suspended, the analysis of entailments expressed in the definition of augmentation would still be problematic, for a more technical reason. According to the definition given, the inferences from (44a) to (44b), (44c) and (44d) are encoded in the semantics:

(44a) Sue was singing on her birthday with Mary in the shower.
(44b) Sue was singing on her birthday with Mary.
(44c) Sue was singing on her birthday.
(44d) Sue was singing.

However, the rule as it stands does not express the augmentation of the unaugmented, initial predicate with the second and third adjunct, or the augmentation of the initially augmented predicate with the third adjunct[80]. This means that the following inferences are not encoded in the semantic representation of a sentence like (44a):

(45a) Sue was singing with Mary.
(45b) Sue was singing in the shower.
(45c) Sue was singing with Mary in the shower.
(45d) Sue was singing on her birthday in the shower.

On the assumption that the inferences in (45) are as valid as those in (44), there is no reason for not including them in the semantics. But it is not clear if the

80 It should be noted that in McConnell-Ginet's (1982) proposal Augmentation is not used recursively, so that this problem does not arise.

rule could be modified to achieve this, since it would imply that already established augmentations have to be undone in order to recover the initial unaugmented verb. Even if such a rule could be formulated, it would complicate the interpretation process rather unduly. Thus, for both theoretical and technical reasons, it seems better to take the entailments out of the semantic rule for augmentation.

The modification of the definition of admissible augmentation to exclude entailments results in the version given below:

(46) *Admissible Augmentation for e* (final version)*

> Let α be a verb of Ty(e* –> X) that translates into an n-order predicate denoting an n-ary relation $\mathbb{R}$. Then α^+ is an admissible augmentation of α with Ty(e –> X) only if α^+ translates into an $n+1$-ary relation denoting $\mathbb{S}$, where $\mathbb{S} \leq \mathbb{R} \times$ Ty(e). The augmented verb α^+ is admissible relative to $\xi \in$ Ty(e) only if $\mathbb{S} \leq \mathbb{R} \times \text{Den}(\xi) \neq \emptyset$.

The changed rule defines the denotation of the augmented verb as the set resulting from the product of the original verb's denotation and the denotation of the adjunct. The e* semantic rule does not need to be changed since it now simply refers to the new rule of admissible augmentation. A corresponding derivation would proceed as the one presented above, but with only one lambda term for the formula value of the verb at each stage, which is increasingly 'enlarged'. The effect of the rule for the example discussed in the sample derivation above is that with the new version of augmentation, the formula value holding at Tn(0) after the application of Elimination would be (47a) rather than (47b):

(47a) Fo(sing^{++}(in_the_shower')(with_mary')(sue'))

(47b) Fo(sing(sue') & sing^{+}(with_mary')(sue') &
sing^{++}(in_the_shower')(with_mary')(sue'))

With this version, the rule licenses the augmentation of a predicate of a given arity which results in an augmented predicate where the arity has been increased by one. This slimmed version of augmentation takes away some of the insights expressed in McConnell-Ginet's original rule. In effect, all that it says is that an optional argument is interpreted if an n-ary verb can be interpreted as an n+1-ary verb in a given model. In the next section, I turn to the question of the relevant models for this semantics, and the role of the denotation of optional Ty(e) expressions employed in the definition of admissible augmentation.

3.3.3. Admissibility

The last notion to be discussed in this section is the notion of admissibility. As discussed above, admissibility can not be defined by the categorial range of the optional Ty(e) expression, which are of the same type as arguments and do not contribute any thematic restriction. Admissibility is thus restricted only by the requirement that the augmented predicate can be asigned a denotation in a given model, that is, that a predicate with the appropriate arity is defined. So, for example, for the interpretation of the example sentence in the derivation above, a corresponding ternary predicate Sing<x, y, z> has to be defined in the model, otherwise the augmentation is not allowed, and no semantic interpretation can be assigned to the string. Given the semantics as defined here, the existence of an appropriate predicate in the model is tested at every application of augmentation, according to the assumption that syntactic operations are matched by corresponding semantic operations. However, the overall result would be the same if augmentation was defined as applying only after all input was scanned. The semantic rules for e^* defined in this section then can be seen as the model-theoretic counterpart of the definition of e^* proposed in the last chapter. Both syntactically and semantically, underspecified verbs derive predicates of varying arities. For a model-theoretic interpretation, this implies that predicates of varying arity have to be defined in the model against which the string is evaluated.

However, from the perspective of LDNSL, model theoretic interpretation is not assigned to the string directly. Rather, hearers build structured mental representations of content, corresponding to the proposition expressed by the utterance. During the process of structure building, hearers employ inferential abilities to derive the maximally relevant interpretation of the words provided. Thus the interpretation of underspecified verbs is achieved at the level of propositional form and is established with recourse to both the syntactic rules of the LDSNL model and freely available pragmatic processes contributing to the establishment of meaning in communication. Since, in this view, natural language is interpreted in conceptual structure, the question of denotations in a model is of secondary importance. In the following chapter, I thus turn to the question of how underspecified verbs contribute to the establishment of the proposition expressed.

4. Summary and Conclusion

In this chapter I have addressed the question of how the underspecified verbs of type e* can be semantically interpreted. This involved the discussion of proposals from the literature, and how these proposals relate to the perspective adopted in LDSNL. Three different approaches to the semantic analysis of verb modification have been discussed in detail; the proposal to treat verb modifiers as functors which take verb phrases as arguments as developed in Dowty (1979), the alternative analysis of McConell-Ginet (1982) where modifiers are treated as arguments of augmented verbs, and Minimal Recursion Semantics (Copestake et al. 1997) which provides underspecified semantic representations where predicate–argument structures can be semantically represented as scope dependencies. After discussing these analyses, I have developed a semantic analysis of underspecified verbs based on the analysis provided by McConnell-Ginet. The modified incremental transition rule e* Partial Resolution in conjunction with the appropriately adjusted definitions of semantic interpretation of e* and admissible augmentation provide a means to assign model theoretic interpretation to underspecified verbs. However, after discussing the analyis, the conclusion was drawn that the interpretation of underspecified verbs can be better characterized as their contribution to the establishment of the proposition expressed. Pragmatic processes relevant for the interpretation of e* will accordingly be discussed in the next chapter.

Chapter 5

Concepts

1. Introduction

In the last chapter, I have explored several ways to formulate semantic rules for underspecified verbs. The formulation proposed provides a means to extend the arity of the predicate denoted by the verb, so that the introduction of a Ty(e) expression in the syntax is matched by the introduction of an argument slot in the semantics. Based on McConnell-Ginet (1982), I have proposed a rule where this process of argument adding is restricted by a notion of admissible augmentation. However, what does count as an admissible augmentation was left rather unspecified, especially since no 'thematic' information has been formally expressed. In effect, the rule relies on the assumption that predicates of varying arity are defined in a given model, which would for example have to include predicates like sing'(x), sing'(x, y), sing'(x, y, z), etc. The background here is that I have in Chapter 4 assumed that natural language expressions are mapped directly onto some model-theoretic interpretation. In this chapter, I drop this assumption, and discuss the Relevance Theoretic distinction between 'logical form' and 'propositional form' and how this distinction is interpreted in the LDSNL model (Section 2). Against this background, I introduce the notion of mental concept and the process of concept formation. The interpretation of e* can with the aid of these notions be analysed as an instruction to concept formation, which includes the enrichment of the predicate, and plays a role in context selection (Section 3). Section 4 concludes the theoretical part of the thesis and summarises the syntax, semantics, and pragmatics of verbal underspecification.

2. Theoretical Background

The development of the arguments discussed in this dissertation so far reflects to some extent the major positions on knowledge of language found in the literature (cf. Kempson 1988a). Although e* is formulated within LDSNL, I have progressively focused on assumptions shared by alternative approaches – the formalization of e* as a syntactic rule was based mainly on syntactic evidence, within a restrictive model of tree building. Conceptually, this corresponds to Chomsky's position that the grammar of a language be specified by a set of psychologically plausible rules which generate ideally all and only the well-formed strings of a language[81]. Chapter 4 then introduced the further assumption that natural language expressions are systematically related to model-theoretic interpretation, an assumption not held by Chomsky, but characteristic of formal grammar approaches following Montague. The missing assumption to distinguish LDSNL from both Generative and Formal Grammar is finally provided here, namely that knowledge of language, including syntactic and truth-conditional semantic aspects, is embedded in a representational theory of mind. This assumption links LDSNL to Fodorian (1981) cognitive philosophy and to the cognitive psychology developed in Relevance Theory (RT) (Sperber & Wilson 1986, 1995).

As has been discussed in Chapter 1 of this thesis, Relevance Theory assumes that interpretation of natural language sentences is established in conceptual structure. The interpretation of an utterance includes its associated propositional form, a mental repesentation, which is established with recourse to pragmatic, conceptual reasoning. As was pointed out earlier, pragmatic processes in utterance interpretation include reference assignment, disambiguation and conceptual enrichment. One reason for this representational assumption is the indexicality of a number of natural language expressions. Most prominently pronouns cannot be evaluated per se in a model, but their denotational content needs to be fixed by a value provided from the linguistic or non-linguistic context. Secondly, ambiguous lexical items are disambiguated in context. These two processes, reference assignment and disambiguation, are standardly assumed to be pragmatic in nature, yet to apply before a natural language string can be truth theoretically evaluated (cf. Carston 1988). In RT terms, the logical form, that is, the output of the grammar, needs to be pragmatically developed into a propositional form, which is the representation capable of truth theoretic evaluation (Sperber & Wilson 1986/1995).

But just reference assignment and disambiguation do not seem to be quite enough. Carston (1988) takes the English conjunction *and* as an example and shows that *pragmatic enrichment* plays an equally necessary role in establishing the propositional form of an utterance. It is uncontentious that

81 Corresponds to some extent, that is. Psychologically plausible for Chomsky means that the grammar should reflect the fact that language is acquired, rather than learned, while for LDSNL psychologically plausible means that language is used by hearers to arrive at an interpretation. Furthermore, current Chomskian linguistics would not postulate 'rules', but rather constraints. Yet, at the level of syntactic analysis Generative Grammar and LDSNL share a number of assumptions.

and on occasion means more than denoted by the corresponding logical connective. For example in (1) *and* has a reading where the second action follows the first (and probably also a causal relation):

(1) She became an alcoholic and her husband left her.

The question for the temporal connotation of *and* in (1) is whether this meaning is an implicature, that is, an inference from the literal, truth conditional meaning of *and*, which is independent of this inference, or whether the temporal meaning contributes to the truth conditions of (1). Carston (1988) argues that the latter is the case. Technically, the temporal meaning of *and* in (1) is an explicature, a case of pragmatic enrichment, which has to be derived in order to establish the propositional form of (1). That this is so can be seen from the interaction of explicatures with logical operators such as negation and disjunction (Carston 1988: 172/173):

(2) It is not the case that she became an alcoholic and her husband left her, but rather that her husband left her and she became an alcoholic.

(3) Either she became an alcoholic and her husband left her or he left her and she became an alcoholic; I'm not sure which.

If the temporal/causal meaning of *and* was merely an implicature, then (2) should expess a contradiction (of the form $(\neg(P \& Q) \& (Q \& P)))$, and (3) a tautology (of the form $((P \& Q) \vee (Q \& P)))$, which is intuitively not true. In contrast, the meaning of (2) and (3) can be adequately explained if it is assumed that the temporal/causal meanings of *and* are explicatures which do contribute to the propositional form, since under this view it is those enriched propositional forms which can be meaningfully negated/disjoint in (2) and (3). Enrichment is thus, in addition to reference assignment and disambiguation, a pragmatic process which bridges the gap from logical form to propositional form.

Carston further argues that the proper theory to express this observation is Relevance Theory. Given that RT is a psychological theory, enrichment is a psychological process, in particular a non-demonstrative inferential process performed by the hearer and guided by the communicative principle of relevance[82]. This ensures that there is both a lower limit and an upper limit

82 See Chapter 1 for the discussion of these concepts.

for the derivation of explicatures. The hearer is expected and entitled to derive as many explicatures as is justified given processing costs. In effect, the hearer infers enough to get a truth evaluable relevant propositional form and then stops.

Relevance Theory is intrinsically connected to the Fodorian concepts of mental representations and the language of thought (Fodor 1975, 1981). The linguistic module is taken as an input module which derives uninterpreted structures. These structures provide the input to the general reasoning faculty which 'translates' logical forms into the language of thought. Cognitive effects, the result of processing an utterance, are computed in the language of thought. They include the establishment of new conclusions from interacting assumptions, the derivation of contradictions and the subsequent abandoning of previously held assumptions, and the confirmation of existing assumptions (Carston 1988: 168). The language of thought is thus inferential. Furthermore, given that truth conditions cannot be checked against a logical form, but only against an enriched propositional form, there is no direct relation between natural language and objects in the world. Rather, this relation holds between objects in the world and propositions, i.e. mental representations in the language of thought.

The LDSNL perspective on mental representations is conceptually very similar to the RT stance, but differs in detail. The difference concerns the notions of modularity and encapsulation. RT is, at least in principle[83], committed to the Chomskian view of the language faculty as an encapsulated module (cf. Fodor 1981, Chomsky 1995). I take this here to mean 1) that the language faculty can be characterized as a computational system with rules and operations specific to language, not found elsewhere in the mind (in that sense, it is a proper module), and 2) that it can exclusively be so specified, that is, not only are there language specific rules, but also no other mental specifications (i.e. rules, operations, format of representations) have access to the language faculty – the only interaction between the language faculty and other mental faculties obtains at (syntactically) designated interface levels (in that sense, it is encapsulated). In contrast, in LDSNL, the language faculty is not

83 I am hedging here since, despite the overall conception, there is little (no?) work in RT which models precisely the interaction between pragmatics and GB/MP LF configurations. In practice, it seems to me that RT just runs off S-structure. Yet this is impossible given a sufficiently strong interpretation of encapsulation (such as I formulate in the next sentence) under which S-structure is a language faculty internal level of representation (or, in MP, not existent at all) and as such precisely not an interface level. Conversely, models of language where pragmatics or world-knowledge can interact with S-structure (or the respective corresponding level(s)) are not encapsulated in the relevant sense.

encapsulated, i.e. 1) holds, while 2) does not hold. In particular, pragmatic reasoning – non demonstrative inference – applies during the process of tree construction (Kempson 1996). The output of the phonological parser is taken to build propositional forms directly, by an interaction of general reasoning and structure building rules. Although the language faculty can still be viewed as a Fodorian module (the rules are module specific), it cannot be characterized as encapsulated (there is no designated output, or interface level)[84].

The view that structure building and pragmatic interpretation interact freely in utterance interpretation is motivated by the problem of anaphoric interpretation. A unified analysis of pronominal interpretation requires, according to the LDSNL position, both a representational conception of natural language, and the interaction of structure and pragmatics (Kempson 1996, Kempson et al. 1999)[85]. The problem is that the denotational content of pronominal expressions, in addition to their general context dependency, varies between different 'readings', i.e. apparently different discrete semantic requirements. Cases in point include bound-variable (4a) and E-type readings (4b), cross-sentential cross-reference (4c), and bridging cross-reference (4d) (examples from Kempson et al. 1999: 10):

(4a) Every girl worries that she might get pregnant.
(4b) Most girls passed with distinction. They had worked very hard.
(4c) Sue came in. She was pregnant.
(4d) The Smiths are very nice. He is a doctor.

---

84 Kay (1995) employs a different notion of module. Whereas I have characterized the difference between GB/MP and LDSNL as resulting from the notion of encapsulation, Kay invokes modularity per se. In his conception, GB/MP is modular, while Montague Grammar, Categorial Grammar, HPSG, and the model he describes in the article, Construction Grammar, are non-modular, because each grammatical element is associated not only with its formal properties, but also with the semantic and pragmatic information it 'encodes'. I think he is wrong ontwo counts here (even granting that one can apply the psychological notion of modularity to formal approaches like HPSG without further discussion); 1) syntactic information associated with English words or constructions seems to be domain specific information, irrespective of whatever else is encoded, hence the weaker notion of module (as opposed to encapsulated module) applies, 2) in Kay's conception, non-modularity results from the fact that information is encoded (as rules) which is regarded as 'pragmatic' (at least by Kay himself, presumably). But the interesting distinction here is of course not what one would think of as pragmatic, but, as is well-known from RT, the one between encoding and (non-demonstrative) inference. I think what Kay wants to argue is that conceptual and pragmatic operations interact freely with language specific information, that is, not, or not only, at a syntactic interface level. Again, to me, that is a non-encapsulation, but not a non-modular conception.

85 I only give a brief summary of the argument here. For full discussion and more examples, cf. the references. Note also that pronominal construal is only the paradigm case, but that the same problem of context dependency extends to definite NPs and demonstratives.

Furthermore, pronouns can be interpreted as discourse based anaphorical. This involves the establishment of a relevant context in which the pronoun can be uniquely identified. However, a purely semantic account of context selection (as in e.g. Situation Semantics, Barwise & Perry 1983) fails to identify the intended referent in a discourse situation with two boys and two dogs, and one of the boys holding one of the dogs too tightly (example from Kempson et al. 1999: 13):

(5) He's holding it too tightly.

The problem can be overcome once a level of syntactic representation is assumed, at which pronominal expressions are projected and interpreted with reference to pragmatic reasoning. This is perhaps even more clear with cases of indirect reference, as illustrated in the following example (from Kempson et al. 1999: 14):

(6) John had a heart attack right outside the hospital, and they refused to treat him without an insurance card.

There is no appropriate antecedent for the pronominal expression in the second clause. The interpretation of *they* rather involves the inferential steps that doctors work in hospitals, and that doctors treat heart attacks.

The conclusion from these and related data is that a purely semantic account of pronominal interpretation – be it situations, dynamic binding (Groenendijk & Stokhof 1991, Chierchia 1995), or discourse representation (Kamp & Reyle 1993)[86] – fails to provide a unified characterization of anaphoric expressions. In addition, semantic accounts fail by the nature of the enterprise to address questions of syntactic restrictions on pronouns, such as resumptive pronouns and island constraints. The LDSNL solution to this problem is to postulate a representational level of structure building where interpretation is subject to considerations of relevance. I assume here that this conception is correct, and, furthermore, that it provides the most natural basis for understanding the interpretation of underspecified verbs, which crucially involves the pragmatic process of conceptual enrichment.

---

86 DRT, in contrast to the two other approaches, does postulate a level of discourse representation, and is thus closest to LDSNL. However, in DRS this intermediate level is purely semantic and does not incorporate syntactic structure, which is assumed to be independent (the relation being established by translation rules).

3. From Enrichment to Concepts

A central consequence of the Relevance conception of communication is that inferential reasoning plays a pervasive role in utterance interpretation, and that a code-model is insufficient for the explanation of natural language meaning. Correspondingly, pragmatic enrichment in the development of the proposition expresssed is the rule, rather than the exception in communication. Enrichment does not only apply to 'logical' words such as *and* in the example in the last section. Another example includes (Carston 1988: 158):

(7) She gave him her key and he opened the door.

The utterance in (7) is to be understood such that whoever he is used the key she (whoever she is) gave him to open the door. To derive the proposition expressed for (7), the hearer has to fill in the pronouns with some suitable representations and enrich *and* in the manner outlined above so as to get a stronger temporal reading. On the assumption that denials bear on what has been explicated (the propositional form), and not what has been implicated, (8) shows again that the temporal reading of *and* is an explicature (Carston 1988: 172):

(8a) No. He opened the door before she gave him the key.
(8b) No. He opened the door and then she gave him the key.

Now for the third process of enrichment, the hearer has to 'fill in' that it was the key she gave him that he used to open the door, to derive a propositional form like (9):

(9) [Ann] gave [Sean] [Ann's] key and[then] [Sean] opened the door [with Anne's key]

The relevant bit here is the PP which is filled in as an explicature. That it is not just an implicature can be seen from the denials in (10):

(10a) No. He had his own key all the way through.
(10b) No. It wasn't locked.

What is being denied in (10) is not that Anne gave her key to Sean, but that Sean used that key to open the door – (10a) says that he used his own key, (10b) says that he didn't use any key at all.

But is the representation in (9) the right way to think about these things? The difference between the *and* case and the *with the key* case is that in the former, we enrich the meaning of the word *and*, whereas in (9) we enrich a formal ('logical') structure. Of course, if we wanted to *encode* the explicature communicated by (7), we would use a PP like the one filled in in (9). But are these two processes independent of each other? On the analogy with the enrichment of *and*, the target of the enrichment in (7) seems at least partly to be the meaning of *open*. In order to state that more clearly, we need the notion of concept.

3.1. Fodorian Concepts

Another notion of the Fodorian theory of mind is 'concept' (cf. especially Fodor 1998). It is this use of concept which is the intended meaning of saying that LDSNL formula values are, or address, concepts. For Fodor, concepts are atomic elements of the language of thought, which combine so that inferences can be stated (assuming that the operations of the language of thought are correctly characterized as inferences). Their main purpose within the Fodorian system is to develop an analysis of lexical meaning, which is, for Fodor, minimal. Words address concepts and concepts combine compositionally at the level of the language of thought. Fodor's point is that there is no lexical semantics; that is, there is no meaningful level of analysis where it can be stated that *kill* 'means' *cause to die*, or that *bachelor* 'means' *unmarried man*[87]. For Fodor, the meaning of *kill* is that it addresses the mental concept kill'. The inferences from (11) to (12) or (13) are inferences over propositions at the language of thought, not linguistic or conceptual inferences:

(11) John killed Bill.
(12) John caused Bill to be dead.
(13) Bill is dead.

The reason for not attributing any 'inferential roles' to lexical items, and therefore, according to Fodor, to concepts, is that it is impossible, so Fodor

87 Fodor's argument is thus directed against the lexical decomposition approach in Generative Semantics, but also against 'structural' linguistic analyses of lexical items as meaning components (cf. e.g. Lyons 1977).

argues, to distinguish exactly which inferential properties are constitutive of word meaning and which are merely encyclopedic facts; or, in other words, which inferences are 'analytic' (hence 'semantic') and which are 'synthetic' (hence 'world knowledge'). From this perspective, any attempt to sort out invariable or necessary parts of word meaning so as to put them into a lexical entry is futile. Furthermore, even if it were possible to find all relevant entailments for a given lexical item and define a word by them (e.g, again *kill* as really meaning something like 'cause to die', and nothing else), the consequence would be that there would only be a word *kill,* but no concept or other mental representation of kill', which would make it impossible to think about killing. In order to avoid this consequence, all words for which entailments can be stated necessarily need to have an associated concept. Thus, the only meaning component of the lexical entry for *kill* is that it addresses the concept kill'[88]. In LDSNL, this is expressed as the formula value, while all other information found in lexical items, e.g. types, control, pointer movement is purely syntactic. However, in contrast to Fodor, LDSNL assumes with Relevance Theory a gap threatening to be a crevasse between linguistic form (words) and mental representation (concept). Fodor's criticism of inferential roles is, from this perspective, valid for lexical items, but not for concepts. The broadly Fodorian view of concepts espoused by LDSNL, and the different conception of natural language content is discussed in the next section.

3.2. Relevance Theory Concepts

Recent work in Relevance Theory has addressed the idea of concepts, and the role concepts play in utterance interpretation (Carston 1996, Sperber & Wilson 1997). Concepts can be thought of as storage points for information. Since concepts are entities at the language of thought, and since operations at the level of the language of thought are inferential, concepts have to interact with assumptions, that is, propositions (S&W 1997). Concepts thus aid in accessing 'chunks of knowledge'. But are there only those chunks identified in the mind which we can access by a word? Sperber & Wilson (1997) argue that this cannot be correct. While lexical decomposition results in a situation where there are fewer concepts than words, Fodor argues that there is a concept for every (lexical) word. Sperber & Wilson take this line of thought further and propose

88 There is in addition psycho-linguistic evidence specifically against deriving *kill* from 'cause to die' reported in Fodor (1973).

that there are many more mental concepts than natural language words. The claim is trivially true, Sperber & Wilson argue, in the case of unstable and ineffable concepts – there are far more perceptual stimuli which we can discriminate, and yet do not have a word for them. Similarly, there is a good possibility that we form individual idiosyncratic concepts, e.g. for a particular kind of pain which annoyingly recurs, without having a word for them. Yet in a more interesting sense the claim is also true for stable, effable concepts. In fact, according to Sperber & Wilson, it follows from the RT assumption that communication involves not only decoding but inference in the construction of meaning.

One of the examples discussed in Sperber & Wilson (1997: 116) is the use of *tired* in the following dialogue:

(14) Peter: Do you want to go to the cinema?
Mary: I'm tired.

Since Mary has committed an act of ostensive communication, and furthermore Peter is entitled to assume that her utterance has some bearing on his question, he is invited to use his inferential abilities to derive a relevant answer. He might go through the following steps of reasoning (1997: 116):

(15a) Mary is tired.

(15b) Mary's being tired is a sufficient reason for her not to want to go to the cinema.

(15c) Mary doesn't want to go to the cinema because she is tired.

Peter uses (15b) as an implicit premise to derive (15c) as an implicit conclusion. It appears that the explicit content in (15a) is complemented by (15b) and (15c) at the implicit level. However, as Sperber & Wilson (1997: 117) point out, that is not quite correct – Peter cannot soundly infer (15b) and (15c) from the 'explicit' content that Mary is tired, because the fact that Mary is tired is not in itself strong enough to warrant the inferential process. Rather, the process of implicit inference must be accompanied by a process of enrichment at the level of explicatures. In order to derive (15c) with (15b), Peter has to enrich the concept of Mary's tiredness, just being tired as such is not enough. Peter can assume that Mary wanted to convey that she is tired enough for not wanting to go the cinema. This is of course a much more specific concept than just being tired, but it the one which is needed to derive the relevant interpretation of Mary's answer. Sperber & Wilson describe the underlying reasoning process as one of parallel adjustment: "expectations of relevance warrant the derivation of specific implicatures, for which the explicit content must be adequately enriched" (1997: 117). In this sense, there are many more concepts than words, since in general, concepts result from a process of enrichment on a given occasion. Complementarily, given the pervasiveness of enrichment in utterance interpretation, the literal direct map from a word to a concept is a rather exotic event, a borderline case of the general process where concepts result from contextual fixing. The Relevance theory characterization of concepts thus differs from the Fodorian conception in that words are taken to be only loosely associated with concepts. The establishment of the actual concepts on which the interpretation of the utterance is built is subject to pragmatic reasoning. But after the process of enrichment, the concept does play a crucial part in inference, and can be described with reference to its inferential role. Thus, Fodor's arguments against inferential roles is correct for lexical items, which provide the hearer with an instruction to built a concept, but not for concepts as constituents of the language of thought. Mental concepts do have inferential roles, but they cannot be stated in the lexicon since words do not in general address concepts directly. As can be seen from the example discussed here, it is in fact exactly inferential roles which are essential for the process of constructing the occasion specific concept, say $\textbf{tired}_{21}$ from the word *tired,* since the process is driven by the derivation of implicatures, which are inferences. With this analysis of words and concepts in mind, I now return to the interpretation of underspecified verbs.

3.3. Concepts Addressed by Underspecified Verbs

With the analysis of concepts discussed in the preceding section, the role of underspecified verbs in interpretation can be characterized more precisely. Consider again the case in (7) (repeated here):

(7) She gave him her key and he opened the door.

I assume that the context for (7) is rich enough that the pronominal expressions can be interpreted, in particular that the people talked about are Keith and Karen. The second clause in (7) then has a representation like (16) before completion:

(16) *Tree for "...* he opened the door

```
                          • {Tn(0), ?Ty(t)}
                                  / \
• {Tn(00), Ty(e), Fo(keith')}        • {Tn(01), ?Ty(e -> t)}
                                              /\
  • {Tn(010), Fo(the_door'), Ty(e)}   • {Tn(011), ?Ty(e -> (e -> t))}


  {Tn(011*), Ty(e* -> (e -> (e -> t))), Fo(open')}
```

The tree in (16) is a semantic tree. More specifically, the tree is a representation of the process of how the hearer establishes the proposition expressed – there is, as a matter of principle, no 'logical form'. So if 'with her key' is an explicature, that is, part of the proposition expressed, it should be assigned a place in the semantic tree[89]:

(17) *Tree for "...* he opened the door.

```
                          • {Tn(0), ?Ty(t)}
                                  / \
  • {Tn(00), Ty(e), Fo(keith')}  • {Tn(01), ?Ty(e -> t)}
                                          /\
  • {Tn(010), Fo(the_door'), Ty(e)}   • {Tn(011), ?Ty(e -> (e -> t))}
                                                /\
  • {Tn(0110), Fo(with((karen')'s key')),     • {Tn(0111),
                                 Ty(e)}         ?Ty(e -> (e -> (e -> t)))}

  {Tn(0111*), Ty(e* -> (e -> (e -> (e -> t)))), Fo(open')}
```

Assuming a context where the hearer derives an interpretation of (7), consistent with the expectation of optimal relevance, which includes the implicatures 'keith didn't have a key for the door', and 'the door was locked', the concept addressed by open has to be enriched in the manner indicated in (17), namely as including the information that the opening involved the use of Karen's key. Given that we know that concepts are enriched, and that we want to model a process of enrichment, what (17) shows is that the concept open' is enriched to opening_(that_particular)_door' with information from the concept addressed by the object, and further to opening_(that_particular)_door_with_(that_particular)_key' by building the explicature into the complex concept. Too many concepts? Well, no, given that there are infinitely more concepts than words, and that concepts are the only entities we want to

89 The genitive here remains unanalysed.

postulate at the language of thought, all these concepts play a role in the interpretation of (7). Upon hearing (7) the hearer is entitled to derive an occasion specific, possibly, in fact probably, complex concept strong enough to be used to derive sound inferences. If necessary for the derivation of implicatures, the final complex concept is established, and it is this complex concept which is being negated in (10).

But now that we know what a 'PP' does which is *not* there, namely to aid in concept formation, it is easy to reconstruct what a PP does which *is* in fact there: it is an instruction for concept formation[90]. The syntactic optionality of PPs (and NPs) behaving otherwise as arguments, which I have modelled as e*, the intuition that VP-modifiers behave semantically as modifying the verb, modelled as admissible augmentation, the 'unexpressed PP' as explicature – it all falls into place nicely once the analysis includes complex concepts to be constructed on the fly. The picture that emerges is that verbs provide an instruction for concept formation in the way all other formula values do. However, in addition to this general underspecification, verbs also explicitly encode that the particular concept to be constructed depends on any further information provided by the speaker[91]. Optional expressions of Ty(e) in the VP, i.e. those introduced via e*, function as constraints on the general instruction of concept formation. Given the syntax and semantics of e*, it follows that the constructed eventual concept which is built into the proposition expressed is indeed of varying arity, as outlined in the preceding chapter. On the further assumption that concepts are storage points for sets of assumptions, Ty(e) expressions can be seen as providing filters on context selection. They aid in constructing the relevant contextual assumptions associated with the eventual constructed concept.

In order to show in detail how this process works, I discuss in the following sections pertinent work in Relevance theory which discusses the

---

90 The reason for starting with 'unencoded' as opposed to encoded NP's/PP's is that these cases are more extensively discussed in the RT literature than the pragmatics of the verb phrase. However, the argument from inferred to encoded enrichment I propose in this section seems sound to me.

91 I don't want to claim that this encoding is exclusive to verbs. I am concentrating on verbs here since verbal underspecification is the topic of this thesis. A natural extension of this work would be to consider other construction types. As just one example, a correspondingly underspecified type for nouns, e.g. Ty(e* -> cn), might fruitfully be employed for adjectival or genitive modification in the noun phrase, particularly given similar processes of enrichment as for example in 'bridging' cases:

(i) I was trying to get out of the car but the door was locked

Enrichment in (i) is needed to construct the car's door from *door* in the second clause. In fact, the case of *key* in the example discussed above is similar.

process of concept formation more fully, so as to provide my final analysis of underspecified verbs in Section 4.

3.4. Processes of Concept Formation

The process of concept formation I have invoked in the preceding section is, as already indicated, based on work in Relevance theory. In the following two sections, I discuss in more detail the process of enrichment as it has been formulated in RT. Firstly, I present another example (from Wilson & Sperber 1999) of how enrichment interacts with inferential effects, and secondly, I discuss (based on Carston 1996) how the process of enrichment can be characterized as interacting with encyclopedic information.

3.4.1. Concept Formation with *eat*

Wilson & Sperber (1999) discuss enrichment in the context of truthfulness, in particular with respect to the question of the role of the 'literal truth' of (the proposition expressed by) an utterance in communication. Wilson & Sperber argue that there is no need for notions like truthfulness or literal truth in communication (as for example expressed by Grice's maxim of quality), but rather that whatever notions of truthfulness would be required can be better explained by the principle of relevance. Wilson & Sperber first point out that even for clear cases of violation of truthfulness in communication, as for example in metaphor or fiction, it is not entirely clear how the non-truthful meaning can be recovered from any postulated literal true meaning. Furthermore, literally false information is much more common than just these obvious cases. Notably loose talk cannot be analysed as any overt suspension or flouting of truthfulness, yet what is said in loose talk is literally false. The solution to this problem, according to Wilson & Sperber, is to substitute the problematic notions of literalness and truthfulness by the well-defined RT notions of explicature and relevance respectively. I am here in particular interested in literalness and explicature[92], which relates to the points raised already above, since, according to Wilson & Sperber (1999), there is no

92 The argument concerning the point of truthfulness runs in broad outline as follows. Wilson & Sperber note Grice's statement that "false information is not an inferior kind of information; it is just not information" (1989: 371). Since hearers have justified expectations of the relevance of an utterance, and since relevance is (can be) achieved by improvement of knowledge, hearers have justified expectations of relevant information, which is, by being information, true information. Hence the notion of truthfulness can (could) be defined by Relevance, which is the more (and only) basic notion. For full details, see Wilson & Sperber (1999: 32).

significant level of literalness in utterance interpretation, because all interpretation involves a process of meaning construction. What is of particular interest is that two examples are discussed in detail. The first one involves the utterance "Holland is flat" in the given context of a planned cycling holiday. The utterance is false if flat is taken literally, but in the context the hearer can enrich the concept flat' to mean something like 'good terrain for cycling since there are no steep hills'. This example, similar to the example involving *and* discussed above, involves the enriching of a concept addressed by a single word, albeit a lexical, rather than a 'logical' word. It is the second example discussed in Wilson & Sperber (1999) which is even more interesting in the present context, since it involves a verb. The context of the example is a situation where Lisa visits her neighbours the Joneses who are just about to have supper. The following dialogue ensues (example from Wilson & Sperber 1999: 15):

(18)	Alan Jones:	Do you want to join us for supper?
	Lisa:	No, thanks. I've eaten.

Lisa's answer "I've eaten" could here be analysed to mean literally something like 'Lisa has eaten something at some point in time within a time span ending at the time of utterance'. But clearly Lisa means something more specific than that, namely that she has eaten that evening, and that she has eaten something which might reasonably be regarded as being equivalent to supper. Wilson & Sperber (1999: 17) sketch how Alan might derive an interpretation of Lisa's utterance which achieves adequate contextual effects. The process is sketched as a table, with the interpretive hypotheses he builds given on the left, the basis for these hypotheses on the right, which I represent in the format given in Wilson & Sperber (1999: 17):

(19a)	Lisa has said to Alan "I have eaten"	Decoding of Lisa's utterance
(19b)	Lisa's utterance is optimally relevant to Alan	Expectation raised by the recognition of Lisa's utterance as a communicative act, and the acceptance of the presumption of relevance it automatically conveys

(19c)	Lisa's utterance achieves relevance by explaining her immediately preceding refusal of Alan's invitation to supper	Expectation raised by (19b) and by the fact that such an explanation would be most relevant at this juncture
(19d)	Having eaten supper on a given evening is a reason for refusing an invitation to have supper that evening	First assumption to come to Alan's mind which, together with other appropriate premises, could satisfy expectation (19c). Accepted as an implicit premise of Lisa's utterance
(19e)	Lisa has eaten supper this evening	First enriched interpretation of Lisa's utterance as decoded in (19a) to come to Alan's mind which, together with (19d), could lead to the satisfaction of (19c). Accepted as Lisa's explicit meaning
(19f)	Lisa does not want to eat supper with us because she has eaten supper this evening	Inferred from (19d) and (19e), satisfying (19c) and accepted as an implicit conclusion of Lisa's utterance

Sperber & Wilson point out that their representation of Alan's thought process in (19) is of course only an approximation – his hypotheses are formulated in the language of thought, so that the English sentences in (19) are just rough paraphrases. Furthermore, the thought processes are not necessarily in the sequence given in (19), but rather, interpretation is carried out on-line, starting before the utterance is over, and "interpretive hypotheses about the explicit and implicit content are developed in parallel, and stabilise when they are mutually adjusted, and jointly adjusted with expectations of relevance" (1999: 17). Given these assumptions, what (19) shows, then, is how by using interpretive hypotheses, Alan derives both the explicit meaning of Lisa's utterance (in (19e)) and the implicit conclusion (19f). Further 'weak' implicatures can be derived from (19f) and further world knowledge, the example given by Wilson & Sperber being 'Lisa might accept an invitation to supper another time'. However, the purely encoded part of the utterance in (19a) is not in itself taken to be the meaning of the utterance at any stage in the interpretation process. One of the claims in Wilson & Sperber (1999) is that, given that pragmatic enrichment is always involved in utterance interpretation, there is no justification for postulating underlying hidden constituents in the encoded part of the utterance; the step from (19a) to (19f) is an inferential process which narrows down the time span encoded by the present perfect, and which provides 'supper' as the relevant 'object' of eating. Therefore, the encoded part of the utterance does not provide any restrictor variable for the temporal interpretation, nor an empty variable in object position. The process of enrichment runs purely on the concepts activated by the words in the string – Alan is "using the concept of eating, which Lisa's words have activated in his mind, and narrowing it down to the concept of eating supper, which helps him to construct a relevant-as-expected interpretation of Lisa's utterance" (1999: 19). The same reasoning applies to other possible hidden constituents, as for example in (20) and (21) (Wilson & Sperber 1999: 19):

(20) "I've often been to their parties, but I've never eaten anything" [there]

(21) "I must wash my hands: I've eaten" [using my hands (rather than, say, being spoon-fed)]

The 'locative' and 'instrument/manner' interpretations indicated in the square brackets are pragmatically derived. There is no motivation for postulating an underlying thematic slot.

The example discussed here shows more clearly than the enrichment of and' or flat', that the enrichment of concepts in context interacts with verbal subcategorization, since in the case of *eat*, similar to the example with *open*, the enrichment involves the postulation of an optional NP. Syntactically and semantically, *eat* in this example can presumably be treated as intransitive, even if this means that the denotation of the verb is not truth-theoretically interpretable by decoding only[93]. Taken literally, eat' in this example would then be undefined, but the force of Wilson & Sperber's argument is that literalness cannot be properly defined in the first place: "However, we have argued that a notion of literalness has no role to play in a theory of language use. All utterances involve a process of meaning construction" (Wilson & Sperber 1999: 29). In this example, the enrichment is conceptually mandatory in order for the hearer to arrive at the proposition expressed, it involves the enrichment of the concept eat' to eating_supper', or in fact to eating_supper_this_evening'.

93 On the assumption that there is only the binary predicate eat' in the model.

Given the interpretation of e* so far, the analysis of this example from the perspective adopted here differs slightly from the analysis given by Wilson & Sperber. While I agree with the view that the enrichment is pragmatic and, in this particular example, mandatory, LDSNL assumptions about syntax and semantic representations lead to a different analysis of the structural representations involved in the enrichment process. For me, the particular enrichment of eating discussed here is an e* case, where enrichment interacts directly with the logical tree derived from the words in the utterance. Recall again that LDSNL trees represent vehicles for interpretation, which are subject to pragmatic processes in general, and might encode the need for, or the possibility of further enrichment. Since underspecified verbs do indeed explicitly encode the possibility of further enrichment, and since the final concept derived in the proposition expressed in this example is the ternary predicate eat$^{27'}$, the enriched constituents are part of the process of eliminating the inherent underspecification, as would be corresponding overt NPs[94]. Thus, with e*, the semantic tree derived for Lisa's uttering 'I've eaten' corresponds to the enriched representation in (19e):

(22) *Tree for* "I've eaten"

• {Tn(0), Fo(((**eat(this_evening))(supper))(lisa)**), Ty(t)}

/ \

• {Tn(00), Fo(**lisa**), Ty(e)} • {Tn(01), Fo((**eat(this_evening))(supper)**), Ty(e -> t)}

/ \

• {Tn(010), Fo(**supper**), Ty(e)} • {Tn(011), Fo(**eat(this_evening)**), Ty(e -> (e -> t))}

/ \

• {Tn(0110), Fo(**this_evening**), Ty(e)} • {Tn(0111), Fo(**eat**), Ty(e -> (e -> (e -> t)))}

The bold face font for formula values in (22) is meant to indicate that these are properly enriched and constructed concepts, rather than instructions for concept formation. It should be borne in mind that these are occasion specific concepts, so that Fo(**eat**) does not mean *the* concept of eating, but rather this particular concept of eating which I have earlier labelled eat$^{27'}$. The tree in (22) reflects the two observations that hearers build semantic trees directly, without S-structure or LF (a basic LDSNL assumption), and that the the tree in (22) is

94 The only difference between overtly expressed and inferred Ty(e) expression being the degree of responsibility taken respectively by speaker and hearer (cf. Carston 1996), further discussed below.

the first enriched interpretation of Lisa's utterance and thus her explicit meaning, as argued by Wilson & Sperber. It also naturally models Wilson & Sperber's argument that literal meaning has no place in utterance interpretation, since all structures prior to the tree in (22) are explicitly incomplete structures (by virtue of e*) and thus there is no way to model literal meaning, which is as it should be. Notice that any differences between the view proposed here and Wilson & Sperber's view is not about pragmatics or enrichment, but about the nature of syntax, since here the link between structure building and pragmatically determined interpretation is made explicit, while the relation is less explicit in Relevance Theory. However, I agree with Sperber & Wilson that there is no need for hidden constituents in the syntax, such as, for example, the thematic slots mentioned with respect to examples (20) and (21) above. It is precisely because I model verbal subcategorization as both underspecified and available for pragmatic enrichment that any ('hidden') thematic requirements from the verb are unnecessary. Further information such as regarding objects, means, location or accompanyment can and will be expressed if and when necessary, that is, when licensed by relevant inferential effects in the process of enrichment. Since this process is context dependent, it can not be stated as a context-independent requirement of subcategorization defined as a particular fixed type specification. Analogously, overtly encoded Ty(e) expressions are aids to the hearer in constructing the requisite concept, and they are interpreted according to the communicative principle of relevance such that the proposition expressed and inferential effects achieve optimal relevance. Implications of this view will be summarized in Section 4, but notice that this characterization gives us what was missing from Chapter 4, namely a general characterization of the interpretation of predicates with varying arity, since, under the assumption that truth-conditional content is not assigned to natural language strings directly, but rather to the mental representation of the proposition expressed, 'verb modification' of Chapter 4, that is, concept construction in this chapter, is, as a case of enrichment, subject to the principle of relevance.

In this section, I have discussed an example of pragmatic enrichment discussed in Wilson & Sperber (1999). I have shown that their argument against the role of literalness in utterance interpretation can naturally be expressed by e* under the LDSNL assumption that words are projected on semantic trees directly. The relevant level of representation in utterance interpretation is the level of propositional form. This representation is derived by hearers by taking both information from words and inferentially derived information together to built a semantic tree corresponding to this proposition.

During the process of building semantic trees, underspecification is resolved, and in the case of underspecified verbs, concepts are formed which do have specific inferential roles and a fixed arity. The process of enrichment discussed illustrates the role of Ty(e) expressions in concept formation and thus provides a natural explanation for the interpretation of e* also in cases where Ty(e) expressions are in fact encoded.

Before finally summarizing these results, I discuss more closely possible processes of enrichment with particular reference to Carston (1996).

3.4.2. Concept Formation and Encyclopedic Information

Carston (1996) discusses the relationship between two processes of contextual concept formation, enrichment and loosening. Enrichment cases include the example discussed in the preceding section, where the lexically encoded meaning of eating is enriched to eating supper. Enrichment is characterized by the fact that the enriched concept entails the lexical concept: the set of attributes of the enriched concept is a subset of the attributes of the lexical concept[95]. With loosening, on the other hand, the relation between the constructed concept and the lexical concept is not as clear. In some cases, the loosening is the opposite of enrichment, in that the attributes of the lexical concept constitute a subset of those of the constructed concept. An example for this is bald' in (23):

(23) I love bald men.

Assuming that the lexical concept bald' refers to total hairlessness, the constructed concept includes all men in such condition, but also men with (sufficiently) little hair. In other cases, however, the constructed concept may exclude some attributes of the lexical concept. In the case of flat' in *Holland is flat*, mentioned above, the lexical meaning of flat is not fully included in the new concept. Another example is the case of metaphoric use in (24) (Carston 1996: 73):

(24) Bill is a bulldozer.

---

95 Carston (1996) formulates these relations as relations between extensions, so that the subset relation may hold between individuals in the extension of the predicate. I use attributes here as a more neutral term without implying a particular semantic analysis of the meaning of concepts.

The constructed metaphorical concept bulldozer' in (24) includes among others, and in contrast to the lexical concept, the attribute human', so that not all attributes of the loosened concept are also attributes of the lexical concept, i.e. there is no subset relation as with the enrichment cases.

Carston (1996) argues that despite this difference, both enrichment and loosening can contribute to the proposition expressed, that both processes are potentially necessary to arrive at an interpretation of the speaker's intended meaning.

With enrichment, as said above, the constructed concept comprises a subset of the attributes of the lexical concept. One of the examples discussed by Carston is the concept bachelor' constructed from the utterance in (25) in a context where the speaker has made it clear that she wants to settle down and have children (1996: 63):

(25) I want to meet some bachelors.

The ad-hoc concept constructed in (25) would include young, beautiful, eligible, unmarried men, but not, for example, the pope. So, everything which qualifies as a constructed bachelor' is also a lexical bachelor', but not vice versa. In this sense optional modification of verbs is a case of enrichment[96], so I concentrate on enrichment cases, rather than loosening cases here.

Taking both loosening and enrichment to be potentially relevant for establishing the proposition expressed offers new perspectives on how these processes can be characterized. One important question in this context is whether enrichment necessarily involves the construction of an ad-hoc concept in the sense of an operation on the attributes of the lexical concept. On the analogy of metaphor, a case of loosening, the answer is no. Carston discusses the ('standard') Relevance theory account of the metonymic and metaphoric predications in (26):

(26a) Maria is a divine voice.

(26b) Maria is a nightingale.

In these cases, it is not necessary to construct a new ad-hoc concept: "the properties whose predication of Maria the speaker endorses can be accessed directly from stored information concerning divine voices and nightingales

96 Cf. the discussion of the semantics of e* in the last chapter, in particular McConnell-Ginet's intuition that an augmented verb retains the extension of the unmodified verb.

[...]. An array of implicatures is thereby constructed and a fully propositional form at the explicit level need never be entertained" (Carston 1996: 83). On the analogy of loosening and enrichment, there is the possibility that some enrichment cases could be analysed in a similar fashion. For example "an utterance of 'John's a bachelor' in the context of a discussion of Mary's desire to get married could implicate that John is heterosexual, youngish, eligible for marriage, etc, without the setting up of a new address/label for the narrowed ad hoc concept *bachelor*'" (1996: 83). This is because bachelor has a rich encyclopedic entry including a number of assumptions which cluster together to delimit a stereotype. This would lead to the possibility that only some cases of enrichment need to be built into the propositional form, while others, like the bachelor case, do not have to be built in, since "the intended interpretation can be derived without them (by an encyclopedic sorting process)" (Carston 1996: 84). However, invoking encyclopedic entries does not necessarily imply that no concept is being built. By assuming that explicatures, like implicatures, can be communicated more or less strongly, the building of ad hoc concepts becomes part of the hearer's responsibilty:

> "Just exactly what concept is the hearer of [...] 'Bill is a bulldozer' expected to construct out of the lexical concept *bulldozer*? The construction process is constrained by the information stored in the individual hearer's encyclopedic entry for *bulldozer* and by his bid for an interpretation consistent with optimal relevance. But this leaves a degree of leeway so that the ad hoc concept actually constructed is to that degree the hearer's responsibilty. ... Explicatures are communicated with varying degrees of strength; a conceptual range is endorsed by the speaker without any specific concept in that range being given full endorsement. ..." (1996: 87).

The process here sketches concept formation as an interplay between encyclopedic information and expectations of relevance. The choice of encyclopedic information is occasion specific and hearer dependent, and thus falls within the range of meaning construction discussed in the preceding section. Before discussing Carston's position with respect to the argument developed here, it should be noted that she includes cases of verb meaning in the range of enrichment processes. Although not discussed in detail, the following example is found (Carston 1996: 63):

(27) Mary cut the cake.

The enrichment here is of the concept addressed by cut: "In the case of [(27)], it is not any old severing of the fibres of the cake that would be communicated in most contexts but rather a particular mode of cutting; comparison with different objects of cutting makes this apparent, for instance *grass, hair, cloth, flesh,* etc." (1996: 63). That is, the meaning of *cut* is subject to the same principles of meaning construal as is the meaning of *bachelor*[97].

The main point of interest in the preceding discussion is that Carston postulates a clear link between enrichment and encyclopedic sorting. If enrichment is viewed purely as a process of narrowing down denotations, the extensional semantics for e* provided in the last chapter would probably be all that is needed. The inclusion of encyclopedic entries, however, opens the possibility of operating with richer meaning representations. It is this latter option which I pursue here, leaving extensional denotations, with reference to the preceding chapter, out of the picture in this chapter.

Intuitively, the argument I explore here is that, for example, *cut* provides the hearer with an instruction to access a number of assumptions stored under the encyclopedic entry cut' (where cut' is an expression of the language of thought), as in (28):

(28a) "Mary cut

At this stage, very little can be constructed, but certain assumptions are 'activated', may be 'Mary spent some time cutting', 'Mary was awake and probably reasonably concentrated', 'Mary used an instrument suitable for cutting', 'Mary was in danger'. All of these assumptions are more or less tentative, although the context (i.e. the expectation to derive contextual effects) even here favours some assumptions over others, say knowing that Mary is a two year old child, rather than a grown-up neighbour. Furthermore, assumptions activated by a concept are more or less relevant for determining truth conditions, which is good since it is not obvious that (28a) has truth conditions[98]. In any case, what I want to avoid here is any detachment between analytic and synthetic truth, since by making concepts a set of world knowledge assumptions, the set should be construed rather liberally. The next piece of information, the concept addressed by the cake, does in principle the same

97 Similarly, Wilson & Sperber (1998: 109) mention examples with *open*, contrasting *open the bottle* with *open the washing machine* and comment: "It seems reasonable to conclude that a word like 'open' is often used to convey a concept that is encoded neither by the words itself nor by the verb phrase 'open X'".

98 Depending of course on the predicates in the model.

thing: it provides access to a number of assumptions, things (we think) we know about cakes, possibly 'cakes are sweet', 'cakes are expensive', 'I don't like cake', 'cakes cause tooth ache':

(28b) "Mary cut the cake"

However, the activated assumptions from *Mary cut* serve as context for the assumptions addressed by *the cake*, so they are here selected according to the principle of relevance with reference to contextual effects derivable from combining them. In a given context, the enrichment of cake' may favour the assumptions that Mary cut a real cake, rather than a blow-up toy one, or that it was ready to serve, rather than deep-frozen. However, the concept addressed has also an effect on the concept addressed by the verb. That is here, it provides further information about which assumptions activated by *cut* are being communicated. Of the ones listed above, 'Mary was in danger' is probably out, while 'Mary was reasonably concentrated' might be maintained. Furthermore, Mary probably used a knife (in particular a knife suitable for cutting cake with) rather than a lawnmower. Of course, without any particular context, this assumption is rather weak[99], but on the assumption that it is potentially communicated, it should be part of the activated assumptions. In particular, I propose that this assumption is part of the assumptions addressed by the constructed concept **cut**, that is, it is part of the enrichment process of *cut*. In this way, the difference between (29a) and (29b) can be characterized as a difference in 'speaker commitment':

(29a) Mary cut the cake.
(29b) Mary cut the cake with a knife.

In both cases, the enriched concept of **cut** includes an assumption that Mary used a knife suitable to cut cakes with[100]. However, in (29b), this assumption is explicitly communicated so that the hearer is completely justified in taking this assumption to be both meant by the speaker, and to be relevant, while in (29a), the responsibility of the hearer is much greater; there is no guarantee that the assumption is relevant (and in most contexts it is probably not), nor a

99 In particular, it is doubtful whether it is an explicature, but recall Carston's distinction between strong and weak explicatures, and her observation that cut' is enriched in the presence of a particular object. This particular enrichment seems plausible to me, cf. the corresponding cases lawnmower, scissors/clippers, long scissors, sharp knife/scalpel for the objects *grass, hair, cloth, flesh.*
100 Which also involves an enrichment of knife' in (24b).

guarantee that it is meant. However, the speaker can assume that the assumption is arrived at (if it is arrived at at all) by 'default'. In other words, if it is not meant, it should be explicitly cancelled:

(30)	Jamie:	Where's the cake?
	Fran:	Mary has just cut it!
	Jamie:	Can I have a piece then, please?
	Fran:	No. She cut it with the lawnmower!

In (30), Jamie can safely assume that Mary cut the cake with a suitable knife, that the cake still exists, and that it is sliced nicely into pieces. However, all these inferential effects are cancelled after Fran's second use of *cut*, this time differently enriched. Under the view proposed here, this results from the hearer's responsibility in meaning construction and the speaker's different means of communicating assumptions. Of course, Fran's just saying that Mary cut the cake is not optimally relevant; she should have forseen that Jamie will run into unjustified cognitive effort (involving a lot of belief revision). But the important and general point is that the lack of relevance results from the particular concept formation, not from, say, untruthfulness[101].

There are, then, two aspects of the interpretation of underspecified verbs. While part of the interpretation process results in the building of a specific concept to be built into the proposition expressed, the second part plays a role in context selection. In particular, an underspecified verb accesses a set, or maybe a number of sets of ('encylopedic') assumptions stored in long-term memory, which may cluster together as prototypes, or 'defaults'. However, since the instruction to create 'verbal' concepts[102] explicitly encodes the possibility that further modifying information might be provided, the access is 'tentative', in that, abstracting away from other contextual information, no assumption is actually maintained as part of the interpretation. The introduction of Ty(e) expressions under e* then provides further assumptions, which act as a filter on the assumptions provided by the instruction from the verb. Since increasing Ty(e) expressions is always a process of enrichment[103], the process of filtering can be viewed as monotonic, since with each step of

101 Which would amount to saying that one can't cut cakes with lawnmowers, but given that one can cut with lawnmowers in general (and grass in particular), this would mean that there are two words cut, one for cakes, one for grass, i.e. a rather unwarranted postulation of ambiguity.
102 That is, a particular species of predicate. Again, the restriction to verbs here is probably too narrow.
103 As assumed both in Relevance (e.g. Carston 1996: 62/63), and by McConell-Ginet (1982) as discussed in the preceding chapter.

enrichment involving the introduction of Ty(e) expressions, a subset of assumptions is selected. The interpretation of underspecified verbs then involves both the enrichment of the concept addressed and its place in the proposition expressed, including the eventual concept's arity, and the selection of assumptions accessed during this process in interaction with other contextual information. Both processes are driven by Relevance considerations and involve non-demonstrative inference. However, the possibility of enrichment is explicitly encoded in the syntactic information provided by the verb.

4. Summary and Discussion

After discussing the syntax and semantics of e* in the preceding chapters, I have in this chapter discussed the role of relevance in utterance interpretation, and introduced the notions of concept and propositional form. The LDSNL position with respect to these notions is that hearers construct semantic, i.e. propositional representations directly, and that pragmatic processes may apply at any stage in the process of tree building. This includes the process of pragmatic enrichment, which plays a pervasive role in the establishment of the particular concepts used to built the propositional form, since lexical words merely project a requirement to construct concepts. From this perspective, the notion of a syntactic level with some notion of literal meaning defined over words as a representation of what is said plays no role in the process of utterance interpretation. Against this background, I have discussed cases of enrichment involving concepts addressed by verbs, and have argued that, to the extent that the enrichment is necessary to derive the proposition expressed, the process has to be modelled as contributing to the tree building process defined in LDSNL, since trees are taken to be representations of the propositional form. By analogy with enriched constituents of the propositional form, I have proposed that constituents in the verb phrase introduced by lexical items fulfill an identical role; they help to construct the concept addressed by the verb. The process of enrichment relevant for e* determines the role of the concept in the propositional form, and provides filters for context selection. In this final section, I discuss some implications and consequences of this analysis, before developing an analysis of applied verbs in Swahili based on this notion of concept formation.

4.1. Sample Derivation

I begin the discussion by providing a sample derivation which now includes the full analysis of verbal underspecification developed here.

In a context where I am just coming home from the shops on a Saturday morning expecting my parents in law to come over for lunch later on, my wife informs me about the whereabouts of our seventeen year old son Trevor:

(31) Trevor is baking in the kitchen.

The relevant tree structure after the introduction of *bake* is as follows:

(32a) *Tree for* "Trevor is baking

```
                    • {Tn(0) ? Ty(t)}
                           / \
• {Tn(00), Fo(trevor),      • {Tn(01) ? Ty(e -> t)}
   Ty(e)}

{Fo(λx bake(x)'), Ty(e* -> (e -> t))}
```

Since the identity of Trevor is salient, the formula value introduced by Trevor, Fo(trevor') can be resolved immediately to Fo(**trevor**). The lexical specification of *bake* introduces the underspecified type specification and the instruction to construct a relevant concept. A set of assumptions is latently activated and tentative interpetive hypotheses can be built with respect to the constructed concept **trevor**. Note that I have not analysed *bake* here as lexically transitive, introducing an object node (with possibly a pronominal Fo($\mathbf{u}_{\text{pro}}$) value)[104]. Nor is it necessary in this example to enrich such a node. Given the context, the inferential effects I derive from (31) concern my need to use the kitchen for sorting the shopping and preparing lunch. But I also know that if Trevor is baking, it probably involves the use of some utensils and some foodstuff, resulting in a fair mess. From this I can start to worry whether I have enough time to clean the kitchen, whether I can find everything I need, whether I can even put my shopping bags somewhere. The question what it is that Trevor is baking, or intends to bake (note that the imperfect leaves the progression of the event open) is irrelevant in this context. However, the next Ty(e) expression is necessary to derive these inferential effects:

---

104 In view of the analysis presented here, the analysis of optional arguments as lexical ambiguity proposed in Chapter 3 is in need of motivation. A brief discussion of lexicalization patterns is offered in the following chapter, but a fully worked out analysis remains to be done.

(32b) *Tree for* "Trevor is baking in the kitchen

• {Tn(0) ? Ty(t)}
/ \
• {Tn(00), Fo(**trevor**), Ty(e)}   • {Tn(01) ? Ty(e –> t)}
/ \
• {Tn(010) Fo(**in_the_kitchen**), Ty(e)}   • {Tn(011) ? Ty(e –> (e –> t))}

{Fo(λyλx bake(y)(x)'), Ty(e* –> (e –> (e –> t)))}

The introduction of the PP leads to the partial resolution of the underspecified type licensed by relevance[105], namely by the fact that in order to achieve optimal relevance, the information from Fo(in_the_kitchen) has to be built into the proposition expressed. In the absence of further input, and in view of the fact that a concept for bake' can be constructed which yields an optimally relevant interpretation for (31), I can take the completed tree in (32c) as the proposition expressed:

(32c) *Tree for* "Trevor is baking in the kitchen"

• {Tn(0), Fo(**bake(in_the_kitchen)(trevor)**), Ty(t)}
/ \
• {Tn(00), Fo(**trevor**), Ty(e)}   • {Tn(01), Fo(λx **bake(in_the_kitchen)(x)**), Ty(e –> t)}
/ \
• {Tn(010) Fo(**in_the_kitchen**), Ty(e)}   • {Tn(011), Fo(λyλx **bake(y)(x)**), Ty(e –> (e –> t))}

The process of enrichment of Fo(bake') affects both the arity of the eventual concept and the particular assumptions taken to be communicated by the eventual concept **bake**, which in this case do not include the object of baking, but only those which are needed to derive the relevant contextual implications, including that the kitchen is dirty. As this derivation shows, in the interpretation of e*, syntactic, semantic, and pragmatic factors conspire in the establishment of the proposition expressed.

105 This licensing might correspond to the notion of admissible augmentation discussed in the last chapter.

4.2. Pragmatic Motivation

The motivation for the syntax and semantics of e*, as detailed in the relevant chapters, follows both from LDSNL assumptions and proposals in the literature. The interpretation of e* as an instruction for concept formation developed in this chapter makes use of the independently provided notions of propositional form, concepts, and enrichment developed in Relevance theory and LDSNL. In this section I offer a brief discussion of the relation between these notions as developed in Relevance theory and the way I have used them for e*.

With respect to overall perspective, the e* analysis is in clear accordance with Relevance theory. The analysis highlights both the role of the hearer in communication and the importance of her inferential abilities in deriving meaning from words. It makes use of the process of parallel adjustment of enrichment and inferential effects, and it characterizes enrichment as a process which affects both the propositional form and context selection. The main difference between what I take to be the standard Relevance position and the LDSNL perspective underlying my proposal is the relation between linguistic knowledge and pragmatic reasoning. For the analysis of e*, I have to assume that these different cognitive abilities are not clearly divided – the process of tree building modelled by e* is directly related to inferential effects. However, I don't think that this perspective makes Relevance theory weaker; little, as far as I can see, really depends on LF, in the same way that little depends on the notion of what is said. Both notions might be important in special social circumstances, but neither of them has much to offer for the understanding of how hearers derive meaning from words. What is important are general structure building processes for natural language input, and principles for the resolution of inherent underspecification, constituting, as it were, virtual conceptual necessity. I think on the whole, there is good pragmatic motivation to think about verbal underspecification in the manner outlined here.

4.3. A Note on the Interpretation of e* and LINK

This section is concerned with the question of formalization, although this question is, albeit from a slightly different perspective, discussed more fully in the Chapter 7. Here I mainly point out that the interpretation of e* cannot be expressed more formally than I have expressed it here. That is, while there are comparatively simple formal statements regarding the syntax of e* (the definition of the type and the corresponding transition rules) and the corresponding lexical actions, the bulk of the explanation is left to the more psychological mode of explanation of Relevance theory. In particular the principle of relevance cannot be formally expressed in the absence of a clear formal definition of cognitive effort and contextual effects, and the calculation of the balance between the two. There is, however, one structural reflex of concept formation, namely the difference between e* and LINK mentioned in Chapter 3. This is the structural expression of which concept is being narrowed. Thus in the ambiguous case in (33), the hearer has to decide which concept is being enriched:

(33) Jane saw the toys from the bathroom

If the PP is introduced by e*, the concept addressed by *see* is narrowed, while the LINK operation provides further information for the identification of *toys*[106]. A similar observation holds for the iteration of identical 'thematic' information as in the following examples[107]:

(34a) Marx was born in a little shed in Trier.
(34b) Marx was born in Trier in Germany.
(34c) Marx was born in Trier in Bonn.
(34d) Marx was born in Trier in a little shed.
(34e) Marx was born in Germany in Trier.

Both (34a) and (34b) are fine since the first PP aids in constructing **born** (which I take to be an active verb here), while the second PP modifies the common noun of the first PP. (34c) is odd since the concept of **born** constructed with the first PP cannot be easily narrowed by incorporating the second PP. Similarly, Trier cannot be easily modified by in Bonn. Both options fail since they clash with the world knowledge information that Trier and Bonn are different towns. (34d) and (34e) are interesting in that they seem to have an 'afterthought' flavour. One possible analysis is that the LINK operation (at least here) requires a subset relation to hold between the modified and the modifier. Construing LINK as a 'reduced relative' here would yield a representation where the unlikely 'Trier is in a little shed' and the wrong (given world knowledge) 'Germany is in Trier' are construed. Rejecting this analysis, the hearer might then construe in *a little shed* and *in Trier* as e* cases. The problem with this is that the concept **born** has already been construed as

106 Under the LINK analysis provided by Swinburne (1999).
107 Which are originally I believe from Rodger Kibble. (34b) (and (34e)) are true.

having occured in a location, so that contextual assumptions are taken to include those which are warranted by being born in Trier or Germany respectively. The inclusion of further, more specific information would potentially lead to more inferential effects, so that the context selection at this stage involves an unnecessary step. If this line of reasoning is correct it would explain why (34d) and (34e) sound more odd than (34a) and (34b). However, I do not explore the interaction between LINK and e* any further.

4.5. Free Pragmatic Processes versus Encoding

The final question to be addressed here is possibly the most interesting one. It is the analysis and status of true, obligatory arguments. I have been concerned here with a general explanation of the interpretation of verbs and verb phrases, within which underspecification is a key ingredient. Under the view proposed here, all expressions of Ty(e) in the verb phrase receive the same interpretation, namely as contributing to concept formation. Optionally introduced expressions are licensed by the principle of relevance. Given that this is a completely general and free pragmatic process, fixed subcategorization requirement is from this perspective exceptional, rather than the rule. There are a number of factors in determining the 'obligatoriness' for overt complements, including the saliency of the potential enrichment, as well as the actual predicate:

(35a) ?Everybody was asked to contribute to the departmental fund, so John gave.

(35b) ?Everybody was asked to contribute £10 to the departmental fund, so John gave.

(35c) *?They established a departmental fund and John gave.

(35d) They established a departmental fund and John contributed.

But a number of verbs seem to resist any form of enriched interpretation substituting the overt object(s):

(36a) *Mary walked to the window and closed.
(35a) *Jamie put.
(35b) *Fran set.

Verbs like *close, put* and *set* have to be analysed as requiring two overtly expressed Ty(e) expressions. A technical statement to the effect that some Ty(e) expressions are obligatorily required to be overtly expressed can be formulated with recourse to the lexicon. I have here assumed that constituents of the language of thought have a type value such as, for example, Ty(e). However, given the partly syntactic nature of these types, and given that the surface syntax of natural language differs from the syntax of the language of thought – in particular the notion underspecification rests on this assumption – this assumption might turn out to be too strong. In order to analyse subcategorization, type values might thus be taken as syntactic expressions of tree growth, while the corresponding language of thought types are like natural language types except for the fact that they cannot fulfill subcategorization requirements. Since this difference is, under the view proposed here merely a difference in overt encoding, the types can be distinguished by the fact that syntactic types are provided from the lexicon, whereas language of thought types are provided from general reasoning.

As pointed out above, the fact of subcategorization is surprising for this analysis. The solution sketched here is thus only a technical stipulation without much conceptual content. As for the reason for why there should be subcategorization, I offer two hypotheses. As a first hypothesis, one might think of verbs requiring objects as addressing concepts somehow 'weakly'. Without further information, no concept can be constructed out of the information from *close* alone, even with the aid of general reasoning. But this hypothesis sounds rather stipulatory to me. A second approach would be to think of strict subcategorization as some form of grammaticalization whereby a free pragmatic process becomes more structurally expressed in the grammar, i.e. here the lexicon, so that the general instruction to construct a concept becomes an encoded instruction to construct a concept with recourse to at least one/two Ty(e) expressions. The point is taken up again in the next chapter, where some examples for such a lexicalization process are discussed, but a more comprehensive analysis remains to be formulated. Thus the conclusion is that while the process of concept formation is completely general, strict subcategorization has to be stated as a lexical requirement.

4.6. Conclusion

This chapter concludes the theoretical part of this thesis, in which I have developed an analysis of verbal underspecification within LDSNL, building on the architecture provided by the framework and motivating the analysis by recourse to basic assumptions embraced in LDSNL about the nature of language and the process of utterance interpretation. The analysis shows that underspecification is not only found in the expression of long distance dependencies, but also in the expression of basic predicate–argument structure, at the heart of syntax. Predicate–argument structure is, according to the analysis proposed here, established only at the level of propositional form and involves general inference patterns in the construction of mental concepts guided by the principle of relevance. The analysis thus supports the view that natural language understanding can be characterized as an incremental structure building process which interacts with general reasoning throughout.

The following two chapters explore aspects of the analysis formulated here. Chapter 6 provides an analysis of applied verbs in Swahili based on the notion of concept formation, while Chapter 7 discusses possibilities of implementing aspects of the analysis from the perspective of computational linguistics.

Chapter 6

Applied Verbs in Swahili

1. Introduction

In the preceding chapter, I have discussed how underspecified verbs contribute to the establishment of the proposition expressed in the process of utterance interpretation, and concluded that underspecified verbs encode lexically the possibility for conceptual enrichment. Underspecified verbs address a concept which accesses a set of assumptions. The introduction of optional Ty(e) expressions provides an aid in filtering the assumptions tentatively accessed and in strengthening the concept addressed by the verb. Concept formation is thus related to complementation in that the eventual arity of the particular constructed predicate is established, and that possibly additional Ty(e) expression are incorporated into the predicate. The construction of an occasion specific concept from a lexical concept may thus imply a change in valency, but it does not do so necessarily.

In this section, I review this analysis in the light of new data. I develop an analysis of applied verbs in Swahili which shows that applied verbs provide an instruction to construct an occasion specific concept, in particular a concept that is stronger than that which could have been constructed from the related base verb. In other words, applied verbs instruct the hearer to derive additional contextual effects from the proposition expressed. The argument developed thus crucially involves the hypothesis developed so far, and results in the formulation of an alternative to analyses reported in the literature.

2. Applied Verbs in Swahli: Introduction

Applied verbs are a characteristic of most Bantu languages. They are part of a system of verbal extensions which includes causative, passive, and neutro-passive amongst others. Traditionally, the applicative has been analysed as extending the valency of the base verb so that a new object can be introduced. The effect can be seen in the examples from Swahili in (1)[108]:

108 All data in this chapter have been collected over a number of years in Hamburg, London, and Zanzibar. The following less familiar morphological tags are used: SCCL1 = subject concord class

(1a)	*A-li-andik-a*	*barua*	
	SCCL1-PAST-write-FV	letter	
	'S/he wrote a letter'		

(1b)	*A-li-mw-andik-i-a*	*shangazi*	*barua*
	SCCL1-PAST-OCCL1-write-APPL-FV	aunt	letter
	'S/he wrote a letter to the aunt'		

As can be seen in (1a), the verb *andika,* 'write', is used with one object in its transitive use, but appears with two objects in its applied form in (1b). The examples given in (1) illustrate the prototypical use of applied verbs, namely the introduction of a 'beneficiary' object. Applied verbs can be formed from transitive verbs as in (1) as well as from intranstive verbs as in (2):

(2a)	*A-li-tembe-a*		
	SCCL1-PAST-walk-FV		
	'S/he had a walk'		

(2b)	*A-li-m-tembel-e-a*	*rafiki*	*yake*
	SCCL1-PAST-OCCL1-walk-APPL-FV	friend	his
	'S/he was visiting her/his friend'		

The intransitive verb *tembea,* 'walk, promenade, stroll', when in its applied form can be used with a direct object to mean 'visit'.

The thematic functions of applied verbs include, next to beneficiary, instrument and place[109]:

(3)	*A-li-andik-i-a*	*barua*	*kalamu*
	SCCL1-PAST-write-APPL-FV	letter	pen
	'S/he wrote a letter with a pen'		

(4)	*A-li-andik-i-a*	*barua*	*meza*
	SCCL1-PAST-write-APPL-FV	letter	table
	'S/he wrote a letter on the table'		

However, although the examples in (3) and (4) involve applied verbs, I will argue below that there is a difference in the structure projected from (1) and (2) on the one hand, and (3) and (4) on the other.

1; OCCL1 = object concord class 1; SITU = situational tense; APPL = applicative morpheme; FV = final vowel; the morphological analysis follows Schadberg (1992).

109 Speakers strongly prefer instrument applicatives to locative applicatives, part of the reason for which I try to explain in the analysis proposed here.

Applied verbs are thus morphologically marked verbs which are related to corresponding base forms by a process which traditionally has been described as an increase in valency. The corresponding construction type is referred to as applicative construction, which equally has been characterized as involving a change in valency. A brief summary of previous analyses of applied verbs is given in the next section.

3. Previous Analyses

Analyses of Bantu applicative constructions have been proposed from several theoretical perspectives. The vast majority of these is concerned with benefactive applicatives and the analysis of cross-linguistic extraction patterns. Since I am here concentrating on one language, I discuss a variety of uses of the applied verb in Swahili, so that the emphasis is different than in most previous analyses. I thus discuss previous work only briefly.

Bresnan & Moshi (1993) and Alsina & Mchombo (1993) provide an analysis of applicative constructions, based on data from Chichewa and Kichaga, in the framework of Lexical Functional Grammar. Part of the analysis is concerned with establishing universal hierarchies of thematic roles, while the main point is to establish the correct linking relationships between elements of the LFG levels of representation function and argument structure (f-structure, a-structure). The analysis is lexical, since lexical operations change subcategorization information.

Within the GB/MP framework analyses have been proposed by inter alia Baker (1988), Marantz (1993), Nakamura (1997). The analyses share the idea that applicative constructions involve movement, in particular of the head of an additional constituent (i.e. an 'abstract' preposition) to the head of the verb. Variations of this analysis include incorporation, adjunction, and feature checking. The data discussed are mostly from Chichewa, or from Kimenyi's (1980) work on Kinyarwanda. Although thematic roles are often employed in GB/MP analyses, Marantz (1993) proposes that deep structure (D-structure) is sensitive to richer semantic event structure, so that syntactic projections reflect how speakers construe particular situations.

Finally, Shibatani (1996) develops an analysis of cross-linguistic applicative constructions, including data from Chichewa (i.e. from Alsina & Mchombo 1993), which is formulated in Construction Grammar and is based on the idea that applicative constructions are formed by projecting transitive verbs on a prototypical 'give' construction, which is a language particular lexical schema which encodes both semantic and syntactic information.

Applied verbs can be formed in a given language according to how similar, in the relevant sense, the respective base verbs are to the 'give' construction. Construction grammar is similar to LDSNL in that lexical instructions project rich syntactic structure, but LDSNL does not employ grammatical prototypes or schemata.

In summary, both syntactic and lexical analyses of applied verbs have been proposed from several perspectives. Despite differences in formalization and basic assumptions, almost all previous formal analyses of applied verbs in Bantu agree on the fact that applicative constructions imply an increase in valency vis-a-vis the base verb. In addition, the majority of analyses proposed are illustrated with data originally presented in the LFG or Relational Grammar literature[110].

4. Preliminary Assumptions

As discussed above, previous analyses of the applied verbs in Bantu languages have concentrated largely on examples of the applied verb where a benefactive object is introduced. Furthermore, most analyses assume that the applied function expresses a change in valency, and try to capture the syntactic aspect of this. In this section I discuss and justify some assumptions I make about Swahili verbs, about the representation of the applicative morpheme, and about the status of lexicalized forms of applied verbs, which will be used in the analysis presented in the following section.

4.1. Swahili as e* Language

In this section I present evidence for assuming that the e* analysis developed in Chapter 3 holds for Swahili. This means that verbs encode lexically an underspecified type, and that optional Ty(e) expressions have to be licensed by a preposition or by a locative case marker, in addition to a few cases of adverbial nouns (e.g. *jana*, 'yesterday', *usiku* 'at night').

In general, optional Ty(e) expressions have to be licensed:

(5a)	*A-li-fik-a*	*usiku*	*pamoja*	*na*	*Sudi*
	SCCL1-PAST-arrive-FV	night	with	and	Sudi
	'S/he arrived with Sudi at night'				

110 In addition, traditional grammar analyses are available for many Bantu languages. For Swahili, see e.g. Sacleux (1909), Ashton (1947).

(5b) **Alifika usiku Sudi*
(in the relevant reading; fine as 'Sudi arrived at night')

(6a) *A-li-andik-a barua kwa kalamu nyekundu*
SCCL1-PAST-write-FV letter with pen red
'S/he wrote a letter with a red pen'

(6b) **Aliandika barua kalamu nyekundu*

(7a) *A-li-nunu-a vitabu kwa bei rahisisi soko-ni*
SCCL1-PAST-buy-FV books with price easy market-LOC
'S/he bought books cheaply at the market'

(7b) **Alinunua vitabu bei rahisisi soko*

The example in (5a) shows the modification of the predicate with *usiku,* which is self-licensing[111], and *Sudi,* which is licensed by the complex preposition *pamoja na.* In (6a) *kalamu nyekundu* is licensed by *kwa,* which also licenses *bei rahisi* in (7a). *Soko* in (7a) is licensed by the locative suffix *-ni*[112]. I thus assume that verbs in Swahili are underspecified, and that optional Ty(e) expressions need to be licensed.

A lexical entry for *andika* thus looks like:

(8) *Lexical Entry for* andik

IF ?Ty(e -> t)
THEN make(<d*>), put(Fo(andik'), Ty(e* -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t)),
put(?<d_1> Ty(e -> (e -> t))),
make(<d_0>), put(Ty(e))
ELSE abort

The lexical entry specifies the actions of a transitive e* verb, as outlined in Chapter 3. The formula value gives Fo(andik'), rather than *andika,* since the final *-a* is part of the inflectional morphology[113].

The corresponding tree structure after the actions specified in the lexical entry for *andika* have been performed is given below:

---

111 Nominal adverbials such as *usiku* might be analysed as modifying a time variable introduced with every propositional formula, and thus be LINKed.
112 All of the forms which license Ty(e) expressions are historically nominal; *kwa* and, more transparently, *pamoja* show locative noun classe agreement. The suffix *-ni* is argued by Samsom & Schadeberg (1994) to result from a grammaticalization process with *ini,* 'liver' as source.
113 I continue to use the citation form with *- a* suffix in the running text.

(9) *Tree for* "Aliandik

• {Tn(0), ?Ty(t)}

/\

• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e -> t)}

/ \

• {Tn(010), ?Ty(e) ◊} • {Tn(011), ?Ty(e -> (e -> t)}

{Tn(01*), Fo(andik'), Ty(e* -> (e -> (e -> t)))}

As expected, the verb is unfixed, and the current node is Tn(010). Note that the formula value of the DU at Tn(00) is Fo($\mathbf{u}_{pro}$), indicating that the subject marker in Swahili is pronominal; no overt subject NP need to be encoded. However, if a postposed subject is scanned, the two formula values may be combined by Merge, so that a lexical subject will occupy the subject node[114]. The contribution of the final vowel *-a* relates to tense, mood and polarity, and I ignore it throughout. The scanning of a Ty(e) expression in this situation leads to the fulfillment of the requirement TODO Ty(e) holding at Tn(010) and, in the absence of further lexical input, the tree can be completed by applications of Merge, Completion and Elimination.

I assume that the assumptions made here are correct throughout this chapter. In the next section, I develop a possible analysis of applied verbs within the general outline given here, which follows the assumptions made in previous analyses of applied verbs. An alternative proposal is developed in the following section.

4.2. A Syntactic Analysis of the Applicative Morpheme

First I assume that the applicative morpheme has its own lexical entry, and that it functions to introduce a new node. Such a putative analysis of applied verbs implies that the applicative morpheme functions in a way similar to prepositions and locative morphemes, in that it licenses the introduction of an additional Ty(e) expression.

The following lexical entry for the applicative morpheme *-IL-* reflects this approach[115]:

114 See Bresnan & Mchombo (1986, 1987) for an analysis of Bantu subject and object markers as pronouns.

115 The surface form of the applicative morpheme is subject to phonological processes, one of which is discussed in more detail below; in the abstract form -IL-, -I- indicates a vowel subject to vowel harmony, while -L- indicates an underlying consonant /l/ which surfaces when followed by a historically high vowel; cf. Schadeberg (1992).

(10) *Putative Lexical Entry for* -IL-:

IF	$?Ty(e)$
THEN	$go(\langle u_0 \rangle), make(\langle d_1 \rangle), put(?\langle d_1 \rangle Ty(e \rightarrow (e \rightarrow (e \rightarrow t))), ?\langle d_0 \rangle Ty(e)),$
	$go(\langle u_1 \rangle), go(\langle d_0 \rangle)$
ELSE	$put(?\langle d_1 \rangle Ty(e \rightarrow (e \rightarrow t))),$
	$make(\langle d_0 \rangle), put(?Ty(e))$

The entry distinguishes two cases; if the applicative morpheme follows a transitive verb, the condition will be TODO Ty(e), as shown in the tree above, and the actions in the THEN clause will be performed. If, on the other hand, *-IL-* follows an intransitive verb, the actions under the ELSE statement will be performed. Since the morpheme has a fixed position in the morphological verbal template, it always follows the verbal base, so that if there is no task TODO Ty(e), the applicative morpheme is preceded by an intransitive verb. Ditransitive verbs cannot take the applicative morpheme. Note that here is another advantage of underspecified verbs; without e*, intransitive verbs would fulfill the requirement TODO Ty(e -> t) directly, so that the applicative morpheme could no longer build new argument nodes.

With this lexical definition of the applicative morpheme, the tree in (9) could be continued as (11):

(11) *Tree for* "Aliandiki...

• $\{Tn(0), ?Ty(t)\}$

/\

• $\{Tn(00), Fo(\mathbf{u}_{pro}), Ty(e)\}$ • $\{Tn(01), ?Ty(e \rightarrow t)\}$

/\

• $\{Tn(010), ?Ty(e)\ \Diamond\}$ • $\{Tn(011), ?Ty(e \rightarrow (e \rightarrow t), ?\langle d_1 \rangle Ty(e \rightarrow (e \rightarrow (e \rightarrow t))), ?\langle d_0 \rangle Ty(e)\}$

$\{Tn(01^*), Fo(andik'), Ty(e^* \rightarrow (e \rightarrow (e \rightarrow t)))\}$

The introduction of the two modal statements at Tn(011) results in the building of a new pair of nodes by Prediction after the requirement at the current node has been fulfilled; the predicate *andika* will then turn out to be (minimally) ternary. Whether or not the introduction of the second object is registered at thé unfixed node depends on whether e* Partial Resolution is operative, but since this rule has been introduced mainly for the discussion in Chapter 4, I assume here that e* is resolved by Merge. Note that I have not introduced any thematic information, nor any other requirement on the kind or order of Ty(e) expressions, for the reasons given earlier.

The lexical entry for *-IL-* defined in this section follows the analyses discussed in Section 3 in assuming that the applicative morpheme introduces a new Ty(e) expression, and that it encodes a syntactic operation. Although in the solution sketched here, there is a lexical entry for *-IL-*, the contribution of the morpheme is analysed as being purely syntactic. This follows from the fact that the building of tree structure in LDSNL is to a large extent lexically driven. The crucial point is that *-IL-* is analysed as a separate morpheme with its own particular lexical actions. In contrast, I will develop an analysis below where *-IL-* does not have its own lexical entry, but is found under the entry of the verb with which it is associated.

4.3. Phonological Evidence against a Lexical Entry for *-IL-*

The syntactic analysis of applied verbs sketched in the preceding section was based on the assumption that *-IL-* has a separate lexical entry. However, this assumption is disconfirmed by phonological evidence regarding the domainhood of *-IL-*.

With respect to the overall perspective of the model of utterance interpretation outlined in Chapter 1, the assumption that the applicative morpheme has a separate lexical entry is doubtful. Recall that it was claimed that the role of phonology in utterance interpretation is to divide a continuous input stream into phonological domains to provide lexical access. For the analysis developed here so far, which has been illustrated mainly with examples from English, phonological evidence was not needed. In the case of applied verbs discussed here, however, phonological domains are important, since the phonological evidence indicates that *-IL-* is not a separate domain. The relevant evidence comes from vowel harmony; the quality of the vowel of the applicative morpheme is determined by the vowel in the verbal root:

(12a)	*ham-a*	'move from'	*ham-i-a*	'move to'
(12b)	*fik-a*	'arrive'	*fik-i-a*	'reach, stay'
(12c)	*nuk-a*	'stink'	*nuk-i-a*	'smell nice'
(12d)	*sem-a*	'speak'	*sem-e-a*	'speak about'
(12e)	*omb-a*	'beg'	*omb-e-a*	'ask on behalf'

As can be seen from the data in (12), the applicative morpheme surfaces as [i] after the stem vowels /a/, /i/, and /u/, while after /e/ and /o/, the applicative morpheme surfaces as [e]. The underlying phonological structure of applied verbs can be represented as follows:

(13) *Schematic Representation of Swahili Applied Verbs*

$$
\begin{array}{cccccc}
\mathrm{O} & \mathrm{N} & \mathrm{O} & \mathrm{N} & \mathrm{O} & \mathrm{N} \\
| & | & | & | & | & | \\
X_1 & X_2 & X_3 & X_4 & X_5 & X_6 \\
| & | & | & | & | & | \\
\mathrm{C} & & \mathrm{C} & & () & () \\
 & (\mathrm{A}) & \Longrightarrow & (\mathrm{I}) & &
\end{array}
$$

The first three skeletal positions (X_1 to X_3) correspond to the verbal root which consists of one nuclear (N) and two onset (O) constituents, corresponding to two consonants (C) with an intervening vowel. The applicative morpheme corresponds to the skeletal positions X_4 and X_5, and is lexically associated with a phonological element (I) as nucleus, and an empty expression as onset. The vowel harmony can be analysed as the spreading of the element (A), if present, from the nucleus of the root to the nucleus of the suffix[116]. From the perspective of lexical access, the structure in (13) is evidence for assuming the following phonological domains for applied verbs:

(14) [[CVC] IL]

This is an instance of analytic morphology where both the verbal root [CVC] and the complex form as a whole, that is the applied verb, are phonological domains, but where crucially the suffix is not a separate phonological domain. The phonological evidence thus disconfirms the syntactic analysis sketched above; the applicative morpheme does not provide lexical access because it does not constitute a phonological domain. Rather, from the phonological evidence it follows that information from the applicative is accessed under the entry of the verb with which it is found.

4.4. Lexicalized Forms

In view of the apparently general and uniform syntactic function of the applicative morpheme, the view that it can only be accessed from the entry of the verb with which it combines seems to be surprising. However, given the

116 For a fuller discussion of Government Phonology analyses of Swahili vowel harmony and Bantu vowel harmony more generally, see Cobb (1997), Kula (1997), and Marten (1997).

function of lexical rules in the LDSNL model, an analysis which incorporates the phonological evidence can be modelled, since, as pointed out previously, structure building is largely driven from the lexicon. In fact, an analysis could be developed in which schematic lexical entries for transitive and intransitive verbs are defined which include optionally one of the two clauses given in the entry for *-IL-*, above. While I think that this approach is basically correct, so that I will develop an analysis along these lines in what follows, the perspective that the applicative morpheme is lexically associated with its verb provides a good starting point to consider more data.

There are in Swahili, and in most Bantu languages, a number of lexicalized applied verbs. Although these data have been mentioned, they have not been included in analyses of applicative constructions in Bantu. In fact, Port (1981) argues that analyses of applicative constructions should not include lexical forms, since a generalization would be lost. I believe the opposite is true. For a full understanding of the function of applied verbs, lexicalized forms provide very good evidence, especially in view of the fact that a lexical analysis of applied verbs is supported by independent evidence.

Lexicalized applied verbs are verbs marked with an applicative morpheme, but which fail to introduce a Ty(e) expression. Two different groups can be distinguished.

There are a number of word pairs expressing movement or motion, where the apparent base verb and the applied verb are not distinguished with repsect to their valency:

(15a)	*kimbia*	'run from'	*kimbilia*	'run to'
(15b)	*geuka*	'turn to'	*geukia*	'turn to'
(15c)	*hama*	'move from'	*hamia*	'move to'
(15d)	*nuka*	'stink'	*nukia*	'smell sweet, nice'

The pairs in (15a) to (15c) are transitive, while the pair in (15d) is intransitive. The semantic relation appears to be one of opposition.

Another group of applied verbs has a lexicalized or idiomatic reading under which the difference in valency between base form and applied form disappears:

(16a)	*fika*	'arrive'	*fikia*	'stay'
(16b)	*tembea*	'walk'	*tembelea*	'walk about, promenade'
(16c)	*toka*	'come from'	*tokea*	'come from'
(16d)	*vua*	'take off (clothes)'	*vulia*	'take off clothes'
(16e)	*vaa*	'wear (clothes)'	*valia*	'dress up (in clothes)'

An illustration of the last example is given below:

(17a)

Juma	*a-li-va-a*	*kanzu*
Juma	SCCL1-PAST-wear-FV	kanzu

'Juma was wearing a Kanzu'

(17b)

Juma	*a-li-m-val-i-a*	*mtoto*	*kanzu*
Juma	SCCL1-PAST-OCCL1-wear-APPL-FV	child	kanzu

'Juma was dressing the child in a Kanzu'

(17c)

?Juma	*a-li-val-i-a*	*kanzu*
Juma	SCCL1-PAST-wear-APPL-FV	kanzu

Int.: Juma was wearing a Kanzu

(17d)

Juma	*a-li-val-i-a*	*nguo*	*za*	*rasmi*
Juma	SCCL1-PAST-wear-APPL-FV	clothes	GEN	official

'Juma was dressed up in official/formal clothes'

The sentences in (17) show the difference between the non-idiomatic and the idiomatic readings of *vaa*, 'wear'. In (17a), the transitive base form is used with one object, *kanzu*, a piece of clothing, while in (17b), the applied form is used with the additional benefactive object *mtoto*, 'child'. The sentence in (17c), with the applied form and only one object is unacceptable in the non-idiomatic reading. In contrast, in (17d) the applied form is acceptable under the idiomatic reading with only one object, *nguo za rasmi*, which here enforces the idiomatic reading.

The examples in this section show that some applied verbs can be regarded as fully or partly lexicalized, because they express a specific lexical or idiomatic meaning. In addition the examples considered here appear to be lexicalized because they do not increase the valency of the base form. This state of affairs seems again to favour a lexical over a syntactic analysis; while in a syntactic analysis, both a general rule and the blocking of the rule for particular lexical items in particular contexts have to be stated, in a lexical analysis all information relevant for the interpretation of applied verbs is found in the lexicon. In the next section, I develop a lexical entry for the verb *vaa,* seen in (17).

4.5. Sample Lexical Entry for *vaa*

The lexical entry for *vaa* has to specify a rule for three different cases; the transitive use of the base form, the di-transitive use of the applied form, and the idiomatic transitive use of the applied form. A lexical entry for *vaa* can thus be defined as follows:

(18) *Lexical entry for* va

IF ?Ty(e -> t)

THEN make(<d*>), put(Fo(va'), Ty(e* -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t)),
put(?<d_1> Ty(e -> (e -> t))),
make(<d_0>), put(?Ty(e))

OR make(<d*>), put(Fo(vali^1'),
Ty(e* -> (e -> (e -> (e -> t))))),
go(<u*> ?Ty(e -> t)),
make(<d_1>), put(?<d_1> Ty(e -> (e -> (e -> t))),
?<d_0> Ty(e)),
go(<u_1>),
make(<d_0>) put(?Ty(e))

OR make(<d*>), put(Fo(vali^2'), Ty(e* -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t)),
put(?<d_1> Ty(e -> (e -> t))),
make(<d_0>), put(?Ty(e))

ELSE abort

The complex lexical entry in (14) licenses three different set of lexical actions. The first set comprises the actions needed for the transitive use, the second set is the process performed for the applied form in the non-idiomatic reading. Finally, the third set of actions expresses the idiomatic meaning of the applied form. The two formula values of Fo(vali) distinguish between the two different readings by means of a numerical superscript. Although the entry looks rather long, all three cases have to be covered, which is achieved by the entry. What this entry implies, and I think this is correct, is that three different tree continuations are developed in parallel, one for each set of actions. Recall that the phonological parsing results in two domains; one for the base verb and one for the applied verb, so that both forms are accessed lexically. However, at the time when the lexical entries are scanned, it is not clear which one will eventually be used. In a situation like this, it follows from basic LDSNL assumptions about incrementality as discussed in Chapter 3, that all possible continuations be represented. I discuss this point further below.

4.6. Lexicalization of Relevance

As has been pointed out above, lexicalized applied verbs like the ones discussed here have not received much attention in the literature, on the grounds that these forms are not productive. But that argument does not quite go through. While it is correct that lexicalized forms are not productive, it still needs to be explained, at least in outline, what the underlying pattern of the lexicalization process is. Note that the lexicalized forms are not simple lexical forms, characterized by an arbitrary relation between sound and meaning. Rather, the term lexicalization implies that lexicalized applied verbs are frozen forms which took part in a productive process at some time in the past. This means that one can sensibly ask which productive process has been lexicalized. The problem for an analysis of applicative constructions as valency changing operation is that it predicts that applied verbs should be lexicalized with their appropriate increased valency. But this prediction is, at least on the face of it, not borne out by the data presented here. The problem for valency changing analyses of applied verbs is thus not that lexicalized applied verbs are not productive, but rather that it is not the postulated productive process which appears to have been lexicalized. Proponents of the valency changing analysis are thus forced to either ignore lexicalized applied verbs, or to assume that in the process of lexicalization the main characteristic of the productive process, the additional object, somehow disappeared.

Here I try to develop an alternative hypothesis, which builds on the notion of concept formation. In particular, I explore the idea that applied verbs encode an explicit instruction for concept formation. Under this hypothesis, applied verbs instruct the hearer to construct a concept which is sufficiently different from the concept that would be constructed from the base form in the given circumstances. This does not mean that two concepts are constructed and then compared. Rather, this is a process of strengthening the concept addressed by the base verb plus the instruction that more contextual implications have to be derived. Note that both the base form and the applied form are lexically accessed, so that the set of contextual assumptions addressed by the base from is tentatively entertained also when the applied form is scanned, which, however, adds the instruction to construct a different, 'stronger' concept.

From this perspective, the lexicalized sense of the applied form *valia* can be characterized as follows. Consider again the contrast between the base form and the idiomatic applied form:

(19a)	*Juma*	*a-me-va-a*	*nguo*	*za*	*rasmi*
	Juma	SCCL1-PERF-wear-FV	clothes	GEN	official

'Juma was wearing official clothes'

(19b)	*Juma*	*a-me-val-i-a*	*nguo*	*za*	*rasmi*
	Juma	SCCL1-PERF-wear-APPL-FV	clothes	GEN	official

'Juma was dressed up (wearing official clothes)'

The applied form in (19b) does not differ in valency from the base form in (19a). Rather, two different concepts are addressed. The difference between (19a) and (19b) can be described as follows; in (19a) the hearer is instructed to entertain the proposition that Juma was wearing official clothes, while in (19b), the hearer is instructed not only to entertain the proposition that Juma was wearing offical clothes, but to derive additional contextual effects such as, for example, that he had dressed with great care and effort, possibly had spent a lot of money on the clothes, or maybe that he had a particular reason for wearing the particular official clothes he was wearing. Note that typical uses of the lexicalized applied form *valia* have an object which denotes special clothes, rather than everyday garments, which facilitate the derivation of inferences such as those mentioned above. The difference between (19a) and (19b) is that in (19b) a stronger concept is addressed.

On the assumption that the characterization of the difference between (19a) and (19b) offered here is correct, it appears that the lexicalized form *valia* encodes a particular process of (former) conceptual enrichment vis-a-vis the base form. This means that before the idiomatic sense of *valia* was lexicalized, the form encoded an instruction for concept formation. In the following section, I develop the hypothesis that this is also the function of applied verbs in present day Swahili.

5. Applied Verbs as an Instruction for Concept Formation

In this section, I develop the ideas introduced in the preceding section, and formulate the hypothesis that applied verbs instruct the hearer to strengthen the concept addressed by the base verb so as to derive sufficiently more inferential effects. I show that it is this instruction which is the basic unifying meaning of applied verbs, and that the syntactic facts can be regarded as following from this basic meaning. At the end of the section I propose a formalization of this hypothesis by defining a disjunctive lexical entry, which, however, reflects the conceptual claim only partly. The following section provides an evaluation of the argument presented.

5.1. Pragmatic Licensing

In the discussion of lexicalized applied verbs above, I have noted that the lexicalized form *valia* is prototypically used with an object denoting 'special', rather than ordinary clothes, which, I have proposed, facilitates the derivation of additional contextual effects. A similar observation can be made with respect to the productive use of applied verbs. Consider the following examples:

(20a)	*Salma*	*a-li-ka-a*	*kiti-ni*		
	Salma	SCCL1-PAST-sit-FV	chair-LOC		
	'Salma was sitting on a chair'				
(20b)	*#Salma*	*a-li-kal-i-a*	*kiti*		
	Salma	SCCL1-PAST-sit-APPL-FV	chair		
	'Salma was sitting on a chair'				
(20c)	*Salma*	*a-li-kal-i-a*	*kiti*	*cha*	*uvivu*
	Salma	SCCL1-PAST-sit-APPL-FV	chair	GEN	laziness
	'S/he was sitting on (in) a comfortable chair'				

The example in (20a) shows the intransitive use of *kaa*, 'sit', with a locative marked adjunct. In (20b), the applied form *kalia* is used, which does indeed license the introduction of the unmarked object *kiti*. However, in the present context, the contrast between (20b) and (20c) is the relevant one. Both sentences are grammatical, but (20b) is, in a 'neutral' context, infelicitous, in contrast to (20c), which differs from (20b) in that a more specific object is used. In view of the hypothesis developed here, the interpretation of these data is that an object like chair gives just not enough contextual information to construct, in the absence of any other contextual information, a concept which is sufficiently stronger than the one constructed from *kaa* in (20a), since there is nothing particular about sitting on a chair; it is what one normally does. In contrast, the object in (20c) provides enough information to build a strengthened concept which may license additional contextual effects such as, for example, that Salma was sitting lazily, slumped back, eyes half-closed, that she didn't intend to get up in the near future, that she wasn't nervous. The (weakest) conclusion to be drawn from the example discussed here is that applied verbs not only introduce objects, but also that the use of applied verbs is subject to 'pragmatic licensing'; they need enough contextual information so that a strengthened concept can be constructed.

A similar example is given below:

(21a)	*Bi Sauda*	*a-li-kat-a*	*mkate*	*kwa*	*kisu*
	Ms Sauda	SCCL1-PAST-cut-FV	bread	with	knife
	'Ms Sauda cut bread with a knife'				

(21b)	*#Bi Sauda*	*alikatia*	*mkate*	*kisu*
	Ms Sauda	SCCL1-PAST-cut-APPL-FV	bread	knife
	'Ms Sauda cut bread with a knife'			

The example in (17a) shows the transitive verb *kata*, 'cut' with the object *mkate*, 'bread', and the PP *kwa kisu*, 'with a knife'. The corresponding applied verb in (17b) licenses the introduction of *kisu* as an object, but the sentence is infelicitous in a neutral context. The applied verb can be used for example in a situation where the bread is in some way unsuitable for being cut with knives (e.g. the sort of bread which is broken, not cut, or may be dried, hard bread), or when the state of the sliced bread gives reason to wonder how it was cut. In (21c), proper contextualization is, imperfectly, expressed by adding a demonstrative pronoun to the first object:

(21c)	*Bi Sauda*	*alikatia*	*mkate*	*huo*	*kisu*
	Ms Sauda	SCC1-PAST-cut-APPL-FV	bread	this	knife
	'Ms Sauda cut this bread with a knife'				

Relevant contextual effects might include that Bi Sauda doesn't know how to cut bread properly, or that the kitchen is now full of crumbs, or that the knife is now probably blunt. Given the appropriate context, all those inferences could also be derived from (21a), but the use of the applied verb indicates that some occasion-specific contextual effects need to be derived, so that the use of the applied form in (21b) without sufficient context is inappropriate.

The examples in this section have shown that the use of applied verbs requires the derivation of additional contextual effects, i.e. that the concept constructed from the applied verb be stronger than the concept addressed by the base verb. Since in the examples considered here, the applied verb also licenses the introduction of an additional object, the examples only provide evidence for the conclusion that applied verbs require pragmatic licensing, in addition to their syntactic quality as increasing the valency of the base verb. From the perspective adopted in this thesis, however, this conclusion is suspicious, since it implies that syntax and pragmatics operate independently of each other. Rather, as discussed in the preceding chapter, the introduction of Ty(e) expressions contributes to the process of concept formation. It is thus natural that applied verbs license the introduction of Ty(e) expressions, since they carry an explicit requirement to construct a specific (i.e. strengthened vis-a-vis the base verb) concept, for which additional information from additional Ty(e) expressions might be provided. On the other hand, if a sufficiently strong concept can be constructed without information from additional Ty(e) expressions, they should not be licensed. In other words, applied verbs, in the analysis developed here, should license an additional Ty(e) expression, but they should not require more Ty(e) expressions than (independently) required by the base verb. That is, applied verbs may change the valency of the base verb, but they do not do so necessarily.

5.2. Concept Formation and Valency

In order to show that the hypothesis developed in the last section is plausible, it needs to be shown that productive, non-lexicalized applied verbs do not necessarily require one more object than the corresponding base verb, in particular in contexts where a sufficiently strengthened concept is constructed without the licensing of an additional Ty(e) expression. In this section, I discuss examples which show that this prediction is borne out[117].

Consider the following data:

(22a)	*Bi Asha a-li-andik-a*	*barua*	*kwa*	*kalamu*
	Bi Asha SCCL1-PAST-write-FV	letter	with	pen
	'S/he wrote a letter with a pen'			

(22b)	*Bi Asha a-li-andik-i-a*	*barua*	*kalamu*
	Bi Asha SCCL1-PAST-write-APPL-FV	letter	pen
	'S/he wrote (in) a letter with a pen'		

(22c)	*Bi Asha a-li-andik-i-a*	*barua*	*kwa*	*kalamu*
	Bi Asha SCCL1-PAST-write-APPL-FV	letter	with	pen
	'S/he wrote (in) a letter with a pen'			

The example in (22a) shows the transitive base form *andika*, 'write', with one object and the PP *kwa kalamu*, 'with a pen', while (22b) shows the applied form *andikia* with two objects, so that this sentence is subject to contextual requirements, discussed below. However, the interesting example in the present context is (22c), where the applied form is used as a transitive verb, namely with one object and one PP. It is example (22c) which shows the dependency of valency changing on concept construction; under a syntactic

117 Here and in the following examples, the description of inferential effects is rather impressionistic, as reconstructed from field notes and with the provision that contextual effects are more occasion specific than indicated here. A more formalized analysis of these and similar data with respect to contextual appropriateness remains to be done.

analysis of applied verbs, (22c) should be ungrammatical. Both examples of the applied verb carry a specific instruction for concept formation; if, in a given context, the neccessary additional contextual effects can be derived without the information from additional Ty(e) expressions, no change of valency is necessary, as in the case of (22c). If, on the other hand, contextual effects are derived by including information from another Ty(e) expression, its introduction is licensed, as in (22b). Thus, all examples in (22) instruct the hearer to entertain the proposition that Bi Asha wrote a letter with a pen, but (22b) and (22c) require the derivation of additional contextual effects. In (22b), these might include that Bi Asha did not use a typewriter, or that, given her hand writing, the letter is illegible, while in (22c), possible contextual effects are that the letter is personal, or that, if it is a long letter, her hand must be painful now. Thus, these examples show that, rather than syntactically encoding a change in valency, applied verbs encode an instruction for concept formation, in particular the instruction to strengthen the concept addressed so as to derive more contextual effects than licensed by the base form.

While the example discussed above involved an 'instrumental' Ty(e) expression, the following examples show the same paradigm with an optional 'locative' expression[118]:

(23a)	*Bw Msa*	*a-li-andik-a*	*barua*	*meza-ni*
	Mr Msa	SCCL1-PAST-write-FV	letter	table-LOC
	'Mr. Msa wrote a letter on the table'			
(23b)	*?Bw Msa*	*a-li-andik-i-a*	*barua*	*meza*
	Mr Msa	SCCL1-PAST-write-APPL-FV	letter	table
	'Mr. Msa wrote a letter on the table'			
(23c)	*Bw Msa*	*a-li-andik-i-a*	*barua*	*meza-ni*
	Mr Msa	SCCL1-PAST-write-APPL-FV	letter	table-loc
	'Mr. Msa wrote a letter on the table'			

As in the preceding example, (23a) shows the transitive use of *andika,* here with a locative marked optional Ty(e) expression, *mezani,* 'on the table'. The applied form is used with two objects in (23b); note that the second object, *meza,* is not marked with the locative suffix *-ni.* Again, the last sentence (23c)

118 In general, ditransitive applied verbs are better with instruments than with locatives. Of the examples shown here, (23c), i.e. the transitive use of the applied is strongly preferred to (23b). As discussed here, the acceptability of applied form depends on an appropriate context. In addition, the locative suffix *-ni* might well be bleached, so that its syntactic function is indeterminate. For the present discussion, it is the acceptability of (23c) that is the important point.

is the most interesting one, since the applied verb is used, as the transitive verb, with one object only, and thus provides evidence against a syntactic analysis of applied verbs as encoding a change in valency. The additional inferential effects required by (23b) and (23c) can be characterized analogously to the preceding example; for (23b) they might include that the family had lunch on the floor, or that the table cloth needs changing, while for (23c), they might include that Bw Msa did not, this time, write his dissertation on the table, or that he was writing very concentratedly, or for a long time.

Note that this analysis requires both underspecified verbs and parallel parses, since all Ty(e) expressions are integrated into the verb, and since adjuncts may precede objects[119]:

(23d) *Bw Msa* *a-li-andik-i-a* *meza-ni* *barua*
Mr Msa SCCL1-PAST-write-APPL-FV table-LOC letter
'Mr. Msa wrote a letter on the table'

The example in (23d) is identical to (23c) except for the order of locative adjunct and object. In order to derive the relevant contextual effects, all Ty(e) expressions have to be considered.

As a final example of the relation between concept formation and valency, consider the following examples involving the verb *pika,* 'cook':

(24a) *Mpishi* *a-li-pik-a* *jiko-ni*
cook SCCL1-PAST-cook-FV kitchen-LOC
'The cook was cooking in the kitchen'

(24b) *Mpishi* *a-li-pik-i-a* *jiko-ni*
cook SCCL1-PAST-cook-APPL-FV kitchen-LOC
'The cook was cooking in the kitchen'

In (24a), *pika* is used intransitively with the locative adjunct *jiko-ni,* 'in the kitchen', in the same way as in (24b), where, however, the applied form *pikia* is used. As in the previous examples, the concept constructed from *pikia* has to be stronger than the one constructable from the base form, thus including for example that the cook was cooking extensively, or habitually in the kitchen.

The examples discussed in this section show that concept formation may, but does not necessarily involve an increase in valency. The syntactic effect of applied verbs thus follows from the fact that they encode an instruction for concept strengthening. The data thus also provide evidence against proposals

119 The word order facts are not clear, though. In addition, the corresponding example with an instrumental is unacceptable.

which analyse applied verbs as syntactically increasing the valency of the base verb. Before considering the formal aspects of the analysis developed so far, I discuss applied verbs used with human objects.

5.3. Human Objects

The majority of examples discussed in the preceding sections involved applied verbs with locative and instrumental Ty(e) expressions, but not with benefactives. In contrast, most previous analysis have mainly been concerned with applied verbs with an additional benefactive object. This is, I believe, one of the reasons for why the hypothesis developed here departs from the common assumption that applicative constructions involve an increase in valency, since it is examples like the ones discussed so far which show clearly that applied verbs encode an instruction for concept strengthening, and that syntactic facts follow from that. However, applied verbs with benefactive objects do indeed imply a change in valency, as shown by the following examples (from the beginning of this chapter, repeated here):

(25a) *A-li-andik-a* *barua*
SCCL1-PAST-write-FV letter
'S/he wrote a letter'

(25b) *A-li-mw-andik-i-a* *shangazi* *barua*
SCCL1-PAST-OCCL1-write-APPL-FV aunt letter
'S/he wrote a letter to the aunt'

(25c) **A-li-andik-i-a* *barua* *kwa* *shangazi*
SCCL1-PAST-write-APPL-FV letter to aunt
'S/he wrote a letter to the aunt'

As illustrated in (25), with a benefactive object, applied verbs invariably show an increase in valency with respect to the base form.

For the present analysis the question is then why benefactives should differ from instrumentals and locatives. The first point to be raised in order to explain this fact is the objection to thematic roles, already discussed earlier in this thesis[120]. I have used the notions of locative and instrumental in this chapter as useful descriptive terms, but they did not play a role in the analysis. The contrast can thus not be phrased with reference to the thematic role benefactive. Rather, I believe it is more correct to construe a contrast between human and non-human referents, so that the question really is; why do

120 See Chapter 2, Section 2.2.1., and Chapter 4, Section 3.3.1.

applied verbs with a human object invariably show an increase in valency. Thus phrased, the question can be answered with two statements. First, human objects are invariably morphologically marked on the verb in Swahili; and second, constructing a concept which involves an additional human referent is sufficiently stronger than the concept constructable from the base verb. The first point is discussed in more detail in this section, while the second is briefly discussed in the next section, although I take it to be uncontentious.

Syntactic objects can be morphologically marked by means of a morpheme prefixed to the verb:

(26a) *A-li-on-a* *kisima*
SCCL1-PAST-see-FV well
'S/he saw a well'

(26b) *A-li-ki-on-a* *kisima*
SCCL1-PAST-OCCL7-see-FV well
'S/he saw the well'

The example in (26a) shows the transitive verb *ona*, 'see', with object, but without object marker. In (26b) the object marker is found in the verbal template; *-ki-* agrees in class with the object. The characterization of the difference between (26a) and (26b) as involving definiteness, as implied in the glosses, is at best an oversimplification. However, the point here is that the object marker in examples such as in (26) is generally optional, in contrast to human objects:

(27a) *A-li-mw-on-a* *Sudi*
SCCL1-PAST-OCCL1-see-FV Sudi
'S/he saw Sudi'

(27b) **Aliona Sudi*

(27c) *A-li-mw-on-a*
SCCL1-PAST-OCCL1-see-FV
'S/he saw her/him'

As can be seen from the contrast between (27a) and (27b), with objects denoting human referents, the object marker is required[121]. As can also be seen, the object marker precedes the verb, in contrast to the full NP. Finally object markers can be used as incorporated pronouns, since the sentence in (27c) is

121 See Wald (1993) for an analysis of the role of human and non-human object markers in Swahili.

fine[122]. I assume here that the (human) object marker fulfills a requirement TODO Ty(e), in the same way as the subject marker discussed above. I also assume that Swahili inflectional morphology does provide lexical access, in contrast to derivational morphology, i.e. that the object prefix does have a lexical entry, in contrast to the derivational suffix *-IL-*.

Thus a sample derivation of the sentence in (27a) proceeds as follows:

(28a) *Tree for* "A-li-

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e –> t)}

This parse stage is achieved after subject marker and tense morpheme have been scanned. The next step is the introduction of the object marker:

(28b) *Tree for* "A-li-mw-

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e –> t) $\Diamond$}
/
• {Tn(010), Fo($\mathbf{u}_{pro}$), Ty(e)}

As can seen in (24b), the object marker builds a new argument node which is annotated with a pronominal formula value of Ty(e), and moves the pointer back to the VP node. At this stage, the verb is scanned and introduced at an unfixed node:

(28c) *Tree for* "A-li-mw-on-a

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e –> t)}
/\
• {Tn(010), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(011), ?Ty(e –> (e –> t))}

{Tn(01*), Fo(on'), Ty(e* –> (e –> (e –> t)))}

At this stage, the derivation could end (cf. (27c), above), but there is further input, namely *Ṣudi*. Given the characterization of the (human) object marker as building a Ty(e) node, the corresponding analysis of (human) NPs is that

122 Cf. again Bresnan & Mchombo (1986, 1987).

they are lexically introduced at unfixed nodes with the weak requirement that they be fixed in the tree, that is below Tn(0):

(28d) *Tree for* "A-li-mw-on-a Sudi

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e -> t)}
/\
• {Tn(010), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(011), ?Ty(e -> (e -> t))}

{Tn(01*), Fo(on'), Ty(e* -> (e -> (e -> t)))}
{Tn(0*), Fo(sudi'), Ty(e)}

The unfixed node Tn(0*) can be fixed at Tn(010) by Merge under the further assumption that meta-variables such as Fo($\mathbf{u}_{pro}$) are defined such that they can be replaced by conceptual formula values such as Fo(sudi')[123]. With two applications of Merge, the derivation is thus completed.

The difference between human and non-human objects can thus be characterized by the fact that human objects are introduced into the derivation before the verb, whereas non-human objects are in general introduced after the verb.

From the point of view of concept formation as encoded in applied verbs, this means that at the time the instruction is registered, there is already one potentially optional Ty(e) expression in the tree; all that is required is to check whether a suitably strong concept can be constructed by including an additional human entity into the concept, which is, as I have assumed earlier, generally the case. Thus the implicational relation between applied verbs with human objects can be explained by independent syntactic facts of Swahili, without recourse to thematic roles, and while maintaining the general analysis of applied verbs as encoding an instruction for concept formation.

In the next section I turn to the question how this hypothesis can be more formally expressed.

5.4. Lexical Entry and Sample Derivations

In this section, I develop lexical entries for applied verbs and show the relevant tree transitions.

123 For an analysis of pro-drop language along the lines sketched here, see Kempson, Edwards & Meyer-Viol (1998). I do not go into full details of such an analysis here.

However, the formalization does not fully express the hypothesis developed here. Ideally, following what has been said so far, the valency of a given applied verb would in an obvious way result from the process of concept strengthening. But the way I have defined verbal underspecification and the contribution of verbs to tree building does not offer a way to state this elegantly. In addition, there is no easy way to state the process of concept formation more precisely[124]. As already indicated, I thus assume that lexical entries specify parallel parses, so that conceptual enrichment results effectively in disambiguation. On the other hand, the advantages of a lexical, as opposed to a syntactic approach still hold, so that the formalization proposed in this section is still to be preferred over the syntactic analysis of *-IL-* sketched in Section 4.2.

The lexical entry for *andika* can thus tentatively be defined as follows:

(29) *Lexical entry for* andik *(first version)*

IF ?Ty(e -> t)
THEN make(<d*>), put(Fo(andik'), Ty(e* -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t)),
put(?<d_1> Ty(e -> (e -> t))),
make(<d_0>), put(?Ty(e))
OR make(<d*>), put(Fo(andikIL^1'),
Ty(e* -> (e -> (e -> (e -> t))))),
go(<u*> ?Ty(e -> t)),
make(<d_1>), put(?Ty(e -> (e -> t)),
?<d_1> Ty(e -> (e -> (e -> t)))),
make(<d_0>) put(?Ty(e))
ELSE abort

The entry specifies actions for the base form, which are the actions for transitive verbs. The second set of actions is relevant for applied forms with human objects. Since the argument node of the VP node has already been built and filled, the lexical actions from *andikIL*^{1} need to build the corresponding functor node, and the argument node of that node.

With the lexical specifications given, I consider now a sample derivation of (30) (= (21b), above):

124 In the following chapter, I discuss ways in which this could be done at least in principle.

(30) *A-li-mw-andik-i-a* *shangazi* *barua*
SCCL1-PAST-OCCL1-write-APPL-FV aunt letter
'S/he wrote a letter to the aunt'

In (30), the applied form of *andika*, 'write', is used with an human object in addition to the object of the base form. I assume that the derivation proceeds as in the preceding derivation of *alimwona* up to the introduction of the verb:

(31a) *Tree for* "A-li-mw-

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e –> t)}
/
• {Tn(010), Fo($\mathbf{u}_{pro}$), Ty(e)}

At this stage, the object marker has built the argument node Tn(010). Scanning triggers the lexical statements of the applied form, and the following tree results:

(31b) *Tree for* "A-li-mw-andik-i

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e –> t)}
/\
• {Tn(010), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(011), ?Ty(e –> (e –> t))}
/
{Tn(0110), ?Ty(e) $\Diamond$}

{Tn(01*), Fo(andikIL'), Ty(e* –> (e –> (e –> (e –> t)))}

The next word, *shangazi*, 'aunt', is assigned to an unfixed node, following the analysis outlined above, and the final word *barua*, 'letter', fulfills the requirement TODO Ty(e):

(31c) *Tree for* "A-li-mw-andik-i-a shangazi barua

```
                         • {Tn(0), ?Ty(t)}
                              /\
• {Tn(00), Fo(u_pro), Ty(e)}      • {Tn(01), ?Ty(e -> t)}
                                      /\
   • {Tn(010), Fo(u_pro), Ty(e)}    • {Tn(011), ?Ty(e -> (e -> t))}
                                          /\
          {Tn(0110), Fo(barua'),      {Tn(0111), ?Ty(e -> (e -> (e -> t)))}
                           Ty(e)}
```

{Tn(01*), Fo(andikIL'), Ty(e* -> (e -> (e -> (e -> t))))}
{Tn(0*), Fo(shangazi'), Ty(e)}

As before, two applications of Merge complete the tree.

Next, consider the example in (32) (= (18b), above):

(32) *Bi Asha a-li-andik-i-a* *barua* *kalamu*
SCCL1-PAST-write-APPL-FV letter pen
'Bi Asha wrote a letter with a pen'

The first difference is that there is a lexically overt subject in (32). I assume that the subject is assigned to an unfixed node and merged with the pronominal formula in a similar way as lexically overt objects. With this in mind, the tree at the introduction of the verb looks as follows:

(33a) *Tree for* "Bi Asha a-li-

```
                         • {Tn(0), ?Ty(t)}
                              /\
• {Tn(00), Fo(u_pro), Ty(e)}      • {Tn(01), ?Ty(e -> t)}
```

{Tn(0*), Fo(bi_asha'), Ty(e)}

The lexical actions necessary at this step are not yet defined in the lexical entry above, but I assume that actions from the lexicon result in the following tree:

(33b) *Tree for* "Bi Asha a-li-andik-i-

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e –> t)}
/\
• {Tn(010), ?Ty(e) ◊} • {Tn(011), ?Ty(e –> (e –> t))}

{Tn(0*), Fo(bi_asha'), Ty(e)}
{Tn(01*), Fo(andikIL'), Ty(e* –> (e –> (e –> (e –> t)))}

The introduction of *barua*, 'letter', results in the fulfillment of the requirement at Tn(010), and by applications of Completion and Prediction in the pointer movement to the new functor node. All of these latter actions result ultimately from modal statements introduced from the lexicon. This can be seen when the lexical rules are defined below.

(33c) *Tree for* "Bi Asha a-li-andik-i-

• {Tn(0), ?Ty(t)}
/\
• {Tn(00), Fo($\mathbf{u}_{pro}$), Ty(e)} • {Tn(01), ?Ty(e –> t)}
/\
• {Tn(010), Fo(barua'), Ty(e)} • {Tn(011), ?Ty(e –> (e –> t))}
/ \
• {Tn(0110), ?Ty(e) ◊} • {Tn(0111),
?Ty(e –> (e –> (e –> t)))}

{Tn(0*), Fo(bi_asha'), Ty(e)}
{Tn(01*), Fo(andikIL'), Ty(e* –> (e –> (e –> (e –> t)))}

The final word scanned is then *kalamu*, 'pen', which duly fulfills the requirement holding at Tn(0111), so that the tree can be completed.

As a final example, consider the transitive use of *andikia*, 'write', in (34) (= (18c), above):

(34) *Bi Asha a-li-andik-i-a* *barua* *kwa kalamu*
SCCL1-PAST-write-APPL-FV letter with pen
'S/he wrote (in) a letter with a pen'

I do not give a tree here, since the derivation parallels the derivation of a normal transitive verb. However, the form has to be represented in the lexical entry.

The final lexical entry for *andika* is given below:

(35) *Lexical Entry for* andik *(final version)*

IF ?Ty(e -> t)
THEN make(<d*>), put(Fo(andik'), Ty(e* -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t)),
put(?<d_1> Ty(e -> (e -> t))),
make(<d_0>), put(?Ty(e))
OR make(<d*>), put(Fo(andikIL^1'),
Ty(e* -> (e -> (e -> (e -> t))))),
go(<u*> ?Ty(e -> t)),
make(<d_1>), put(?Ty(e -> (e -> t)),
?<d_1> Ty(e -> (e -> (e -> t)))),
make(<d_0>) put(?Ty(e))
OR make(<d*>), put(Fo(andikIL^2'),
Ty(e* -> (e -> (e -> (e -> t))))),
go(<u*> ?Ty(e -> t)),
make(<d_1>), put(?Ty(e -> (e -> t)), ?<d_0> Ty(e),
?<d_1> Ty(e -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t))
make(<d_0>), put(? Ty(e))
OR make(<d*>), put(Fo(andikIL^3'),
Ty(e* -> (e -> (e -> t)))),
go(<u*> ?Ty(e -> t)),
put(?<d_1> Ty(e -> (e -> t))),
make(<d_0>), put(?Ty(e))

The final version of the lexical entry thus specifies four different sets of actions, for the transitive use of the base form, for the applied form with human object, for the applied form with two non-human objects, and for the transitive use of the applied form, in that order[125]. Note that the first and the last set of actions are identical except for the formula value.

125 There is in addition an intransitive use of the base form:

(i) *ni-li-mw-on-a* *a-ki-andik-a*
1.SG.-PAST-OCCL1-see-FV SCCL1-SITU-write-FV
'I saw her/him reading'

I have ignored this use in the lexical entry above, but it might easily be added.

The formulation of the lexical entry in (35) provides rules for all four readings of *andika* discussed in this section. The first sub-entry can be accessed separately, when the base form of *andika* is introduced into the tree. However, if an applied form is scanned, all four sub-entries are activated. Since the entries specify different tree continuations, four different trees are built. The role of pragmatic inferencing is, under this view, reduced to disambiguation, in a context where there is in fact very little evidence that applied verbs are ambiguous. I have thus chosen to present a descriptively adequate analysis of applied verbs in Swahili, rather than proposing a conceptually more interesting, but, probably less explicit analysis which relies heavily on the notion of concept formation without a clear link between this process and the formal model provided by LDSNL.

6. Summary and Conclusion

In this chapter, I have developed the argument that applied verbs in Swahili encode an explicit instruction to construct an occasion-specific concept which is stronger than a concept constructed from the base form in that more contextual effects need to be derived. Both the process of strengthening and the derivation of additional contextual effects are quite general pragmatic processes and interact freely with contextual information. In order to show that this hypothesis is plausible, I have adduced phonological evidence indicating that the applied verb provides lexical access both to the base form and the applied form, but not to the applicative morpheme. Further evidence was presented from lexicalized senses of applied verbs which show that applied verbs are not necessarily of a higher arity than their corresponding base form. The next step was to show that productive uses of applied verbs which do change the valency of the base verb are in need of pragmatic licensing, which provides a sufficiently rich context to derive additional contextual effects. The final step in the argument was to show that applied verbs can be used without any change in valency provided that an appropriately strong concept can be constructed from the context without the information provided by optional Ty(e) expression. The evidence considered shows that the instruction for concept formation is the basic unified meaning of applied verbs, and that an increase in valency is a possible, but not a necessary syntactic reflex of concept formation. I then have turned to potential counter-evidence in the form of applied verbs used with human objects, which always display a change in valency. I have argued that this fact results from independent syntactic properties of Swahili, namely that human objects are obligatorily marked by an object marker prefixed to the verb. This means that human objects precede, rather than follow the verb, so that they are naturally incorporated into the process of concept formation. The introduction of a representation of a person into a constructed concept provides enough contextual effects for this process to reliably happen. Finally, I have proposed a formalization of this hypothesis in the form of a complex disjunctive lexical entry, but I have noted that the formalization does not express the generalizations captured adequately.

Several conclusions can be drawn from the discussion in this chapter. First, the analysis of the relation between concept formation and valency proposed in this thesis has been supported. Swahili applied verbs constitute evidence for the importance of this relation, since the process of concept formation has been shown to have an effect on complementation. Concept formation may, but does not necessarily, imply a change in complementation, although it does always imply that the eventual arity of a given concept is fixed. The formalization of this relation as underspecified type values for verbs has proven to be correct, although matters of implementation remain outstanding.

The analysis proposed in this section contributes in addition to the study of Swahili and Bantu languages more generally. The argument developed provides an alternative to the view that applicative constructions are sufficiently analysed as encoding a change in valency, which has been proposed on various occasions in the literature. More specifically, the analysis shows that contextual factors need to be addressed even when analysing apparently purely syntactic data. The role of pragmatic licensing, and more generally of contextual assumptions is an area where more research is needed, and I am quite aware of the fact that the treatment here is cursory at best. In more than one respect, the hypothesis developed here needs to be tested against more and more detailed data, not so much, in my view, with respect to grammaticality, but with respect to acceptability. However, I believe that basically, the hypothesis is correct.

Finally, the analysis shows the need to provide a more explicit link between the LDSNL system and pragmatic processes relevant for the construction of meaning in context. The case of Swahili applied verbs illustrates clearly that the process of structure building as modelled in LDSNL is thoroughly intertwined with the process of concept formation explored in Relevance Theory. Yet there is currently no explicit interface to state this relation. While, on the one hand, the model of utterance interpretation developed in LDSNL appears to be too structural to incorporate non-demonstrative inference, the analysis of concept formation in Relevance

Theory is not structural enough to be incorporated into LDSNL trees. From the point of view adopted here, it would be useful if the relation could be stated more explicitly. In the following chapter, I discuss some preliminary steps in that direction.

Chapter 7

Aspects of Implementing Concept Formation

1. Introduction

At the end of the preceding chapter, I have noted that there is no explicit formalism which might be employed to model the link between the process of structure building described in LDSNL, on the one hand, and the pragmatic process of concept formation, on the other. The analysis of applied verbs in Swahili provided a good example for the different aspects of concept formation. Under the analysis proposed in the last chapter, applied verbs encode an instruction for concept formation which may, but does not necessarily, involve a change in valency. Yet it is only this latter effect which is clearly reflected in LDSNL, as a change in type of the verb and the corresponding tree structure. In this chapter I offer a brief discussion of how a framework might be developed which could express the relation between concept formation and valency changing. The overall perspective adopted in this chapter is the perspective of computational linguistics; the question of formalization has thus here also a practical aspect. I introduce and discuss two proposals which model context sensitive inference and reasoning with uncertainty. In particular, work in Generative Lexicon Theory has addressed the question of the context sensitivity of word (including verb) meaning and thus might be employed for articulating the processes of pragmatic strengthening. However, as the following discussion shows, the theory does not provide the necessary resources nor sufficient conceptual clarity to offer a serious possibility for modelling concept construction. I then introduce an alternative to Generative Lexicon Theory developed by Hunter & Marten (1999) which represents world knowledge as a classical logic augmented with default rules, and discuss how this approach could be employed to provide a more explicit link between the LDSNL model and contextual reasoning. However, the discussion is very preliminary, and gives an outline of the system developed, rather than a specific implementation.

2. The Generative Lexicon

Generative Lexicon Theory (GLT) is one recent formulation of lexical knowledge. It is part of a wider trend in (computational) linguistics to analyse lexical knowledge not as an ideally minimal set of knowledge specifying only basic syntactic and maybe thematic information, but rather to think of lexical items as more structured, for example as being related by semantic relations modelled as feature structures, or as specifying certain aspectual information (cf. e.g. Sag & Szabolcsi 1992, Pustejovsky & Boguraev 1996). In this section, I introduce GLT as formulated in Pustejovsky (1995), contrast it with e*, and discuss an extension of the theory proposed in Lascarides, Copestake & Briscoe (1996) and Lascarides & Copestake (1998). This amended version is again evaluated with respect to concept formation as employed in e*.

2.1. Generative Lexicon Theory

GLT as originally formulated by Pustejovsky (1995) proposes that information provided by lexical items is much richer than standardly assumed. In particular, GLT argues that the lexicon of a given natural language cannot simply be characterized as a list of items with only syntactic and minimal semantic information, for three reasons (1995: 39):

1. Words can be used creatively; they assume new senses in novel contexts.
2. Word senses are not atomic; they overlap and refer to other senses of the word.
3. A single word sense can have multiple syntactic realizations.

The creative use of words is found, for example, with the adjective *fast* (1995: 44/45):

(1a)	a fast boat
(1b)	a fast typist
(1c)	a fast book
(1d)	a fast driver
(1e)	a fast decision
(1f)	a fast motorway

The meaning of *fast* varies according to the noun it modifies, thus, Pustejovsky argues, giving rise to at least four different lexical entries (1995: 44/45):

(2a)	**$fast_1$**:	to move quickly
(2b)	**$fast_2$**:	to perform some act quickly
(2c)	**$fast_3$**:	to do something that takes little time
(2d)	**$fast_4$**:	to allow for driving a vehicle quickly on it (ie. the motorway)

There is seemingly no principled end to such a list, nor any way to relate one sense to another. Pustejovsky (1995: 45/46) presents the examples in (3) as being indicative of a similar problem; since a lexical entry for *want* would have to include at least the ones found in (4):

(3a) Mary wants another cigarette.
(3b) Mary wants a beer.
(3c) Mary wants a job.

(4a)	**$want_1$**:	to want to smoke
(4b)	**$want_2$**:	to want to drink
(4c)	**$want_3$**:	to want to have

Again, an increase in NP objects of *want* would lead to an increase in lexical entries (in addition to entries required due to varying syntactic complementation).

The probem of overlapping of different word senses can be seen with *bake*:

(5a) John baked the potatoes.
(5b) John baked a cake.

The verb *bake,* along with *cook* and *fry,* is, according to GLT, ambiguous between a 'change-of-state' and a 'creation' reading; in (5a), the activity of baking changes the state of the object, potatoes, from cold to hot (and not_really_edible to edible), while in (5b), the object is the result of the activity – it did not previously exist. However, according to GLT, these two senses are not clearly distinguishable, even in context, since one sense is included in the other, and since the actual activity is not distinguished by the different objects. Thus, even if one were to allow many different senses, the postulation of two senses for *bake* here would obliviate their partial overlap.

Finally, Pustejovsky observes that a single form can participate in a number of syntactic realizations, corresponding to different senses (1995: 51):

(6a) Madison Avenue is apt to forget that most folks aren't members of the leisure class. (factive reading)

(6b) But like many others who have made the same choice, he forgot to factor one thing into his plans: Caliphobia.
(non-factive reading)

(6c) As for California being a state being run by liberal environmental loonies, let's not forget where Ronald Reagan came from.
(embedded question)

(6d) What about friends who forget the password or never got it?
(concealed question)

(6e) He leaves, forgets his umbrella, and comes back to get it.
(ellipsed non-factive)

By simply postulating several entries for *forget,* both the relation between complement type and reading, as well as the common 'core meaning' of the verb are not expressed.

Thus, Pustejovsky concludes that the only way for a list-lexicon to deal with these phenomena is the postulation of an infinity of different senses for a single lexical item, which is not only unintuitive and cumbersome, but also fails to capture any structure or systematicity between several senses, both semantic and syntactic.

In contrast, GLT designs lexical entries which are structured, and thus can encode different senses, and which allow for the generation of novel senses in composition. That is, the different readings of *bake* (as in (5)) result from the complex lexical entry of the verb and from information encoded in the lexical entry of the object. It is this process of generativity, devising new senses by composition, by which GLT proposes to meet the demands of accounting for creativity, overlap of senses, and multiple syntactic realizations.

In order to achieve this, GLT introduces four levels of lexical semantic representation (1995: 61):

- argument structure
- event structure
- qualia structure
- lexical inheritance structure

Argument structure states in a standard way number and type of logical arguments, with the addition of 'shadow' and 'default'-arguments, different types of semantically necessary, but syntactically optional arguments. Event structure encodes aspectual lexical information, similar to, but more refined than, a simple event-variable. Qualia structure is probably the most novel (and controversial) idea in GLT, and it is also the structure with the most complicated internal structure. Qualia structure values encode information about what the lexical item is (refers to), what it consists of, what it is made of, and what it is used for. The relation among lexical items is encoded in the final structure, lexical inheritance, which expresses hyponymy and other lexical relations on a particular lattice structure. Information from individual lexical items is related to the lexical inheritance structure via the qualia and formal structure values of the lexical item. Similar to HPSG, values of predicates can be co-indexed to indicate feature unification. This particular selection of features is partly motivated by the claim that the information included in the entries is relevant for speakers' knowledge of language; it is claimed to play a role in grammar which distinguishes it from (other) world knowledge[126].

The lexical entry for bake thus looks as in (7) (1995: 123) (shared features are indicated by underlined numerals, rather than the more conventional numerals in boxes found in Pustejovsky):

(7)

$$\begin{bmatrix} \textbf{bake} \\ \text{EVENTSTR} = \begin{bmatrix} E_1 = \mathbf{e_1} : \textbf{process} \\ \text{HEAD} = \mathbf{e_1} \end{bmatrix} \\ \\ \text{ARGSTR} = \begin{bmatrix} \text{ARG1} = \underline{1} \begin{bmatrix} \textbf{animate\_ind} \\ \text{FORMAL} = \textbf{physobj} \end{bmatrix} \\ \text{ARG2} = \underline{2} \begin{bmatrix} \textbf{mass} \\ \text{FORMAL} = \textbf{physobj} \end{bmatrix} \end{bmatrix} \\ \\ \text{QUALIA} = \begin{bmatrix} \textbf{state\_change\_lcp} \\ \text{AGENTIVE} = \textbf{bake\_act}(\mathbf{e_1}, \underline{1}, \underline{2}) \end{bmatrix} \end{bmatrix}$$

The entry illustrates possible values to the three structures of lexical information encoded in lexical entries (the fourth structure, lexical inheritence, serves as ontological backbone which helps interpret the values of the FORMAL parameter). The event structure value identifies 'baking' as a

---

126 Which seems to imply that, unless thinking is construed as being dependent on language, some information is being duplicated since for example the observation that one can eat cakes is part of the lexical meaning of *cake*, but surely we know that independent of the word *cake*, so that both the lexical item and the world knowledge include this statement.

process, the headedness value is exploited for complex event structures and is here of minor importance. Argument structure makes the verb transitive and places further restrictions on the arguments[127]. The qualia structure values, finally, identify *bake* as belonging to the 'change of state' 'lexical-conceptual paradigm' (lcp), over which generalizations over verb-classes can be stated. 'Agentive' means that the (saturated) predicate involves some 'bringing about'. In GLT, this is the only lexical entry for *bake*. The claim is now that objects such as *potatoes* or *carrots* leave the information from *bake* unmodified, but that objects like *cake* 'shift' the reading of *bake* to a resultative or 'create' reading. That is, apparently different senses of the verb really arise from interaction with the lexical specification of (object) NPs, in particular by co-composition of qualia values. The process is triggered from NPs like cake (1995: 123):

(8)

$$\left[\begin{array}{ll} \textbf{cake} & \\ \text{ARGSTR} = & \left[\begin{array}{ll}\text{ARG1} = & \mathbf{x{:}food\_ind} \\ \text{D-ARG1} = & \mathbf{y{:}mass}\end{array}\right] \\ & \\ \text{QUALIA} = & \left[\begin{array}{l} \text{CONST} = \mathbf{y} \\ \text{FORMAL} = \mathbf{x} \\ \text{TELIC} = \mathbf{eat(e_2,z,x)} \\ \text{AGENTIVE} = \mathbf{bake\_act(e_1, w, y)}\end{array}\right]\end{array}\right]$$

The 'D-ARG' (i.e. default argument) indicates that cakes are made from stuff which can optionally be expressed as an oblique argument as 'bake a cake from/with flour'. This is taken up in the qualia structure, where CONST states that cakes consist of (are constituted by) the stuff encodable as a default argument, dough, maybe, or flour, or chocolate (but, not apparently, raisins, which are count). FORMAL encodes most directly what the word actually means, namely food (by feature sharing with ARG1), and this value can be found again at the lexical inheritance structure to give the embedding in the lexical net. TELIC means function, here that one eats cakes, while AGENTIVE indicates how the cake comes into existence, namely by baking. Note that the AGENTIVE value has the same predicate as *bake*, and that the distribution of the variables are

127 There is actually some mismatch here between what is given in the typed feature structure and what, in my reading of the surrounding text, should have been in there; ARG2 should probably be a default or shadow argument to include sentences like 'John was baking in the kitchen'; furthermore, it is unclear why ARG2 is 'mass' here, since, as discussed below, the default reading of *bake* for Pustejovsky is the baking of things like potatoes, hence it should be 'count'. It is my impression in general that given that GLT is a formal account of lexical structure, the formalization is often surprisingly opaque, an impression shared by Fodor & Lepore (1998).

such that the eating event involves an as yet unknown eater (z) and the food variable from ARG1 (x), but that baking involves not (x) in object position, but (y), the variable bound as 'mass' in D-ARG1 – that is, the act of baking does not involve the cake, but rather the stuff out of which it is made. Now the combination of *bake* with *a cake* results in the following semantic representation (1995: 125):

(9)
$$\begin{bmatrix} \textbf{bake a cake} & \\ \text{EVENTSTR} = & \begin{bmatrix} \text{E}_1 = \mathbf{e_1} : \textbf{process} \\ \text{E}_2 = \mathbf{e_2} : \textbf{state} \\ \text{RESTR} = <_\alpha \\ \text{HEAD} = \mathbf{e_1} \end{bmatrix} \\ & \\ \text{ARGSTR} = & \begin{bmatrix} \text{ARG1} = \underline{1} & \begin{bmatrix} \textbf{animate\_ind} \\ \text{FORMAL} = \textbf{physobj} \end{bmatrix} \\ \text{ARG2} = \underline{2} & \begin{bmatrix} \textbf{artifact} \\ \text{CONST} = \underline{3} \\ \text{FORMAL} = \textbf{physobj} \end{bmatrix} \\ \text{D-ARG1} = \underline{3} & \begin{bmatrix} \textbf{material} \\ \text{FORMAL} = \textbf{mass} \end{bmatrix} \end{bmatrix} \\ & \\ \text{QUALIA} = & \begin{bmatrix} \textbf{create\_lcp} \\ \text{FORMAL} = \textbf{exist}(\mathbf{e_2}, \underline{2}) \\ \text{AGENTIVE} = \textbf{bake\_act}(\mathbf{e_1}, \underline{1}, \underline{3}) \end{bmatrix} \end{bmatrix}$$

Amongst other details, the object results in a modified qualia structure which makes the VP (not the verb, incidentally) a member of the 'create' lcp. Furthermore it states that there are two events, one, the AGENTIVE value, involving the baker and the dough, and a second one, the FORMAL value, which is a stative event (cf. event structure value for E_2) at which the second argument, the cake, exists. Of interest are that several semantic intuitions are captured at the interrelated but distinct lexical levels; thus the resulting state of existence can be read off from the event and the qualia structures, while the interplay between argument and qualia structures allows for a mismatch between syntactic and semantic structure – *a cake* is the syntactic direct object, but is not a member of the extension of **bake_act** in the QUALIA.

GLT has highly structured lexical information for members of all parts of speech, and claims with those to provide a principled explanation for different (for GLT, lexical) meanings of words in context, accounting for a large range of readings, as well as for why certain readings are impossible, without the necessity for multiple lexical entries.

2.2. Similarities and Differences between GLT and e*

It is fairly obvious that the GLT conception of the lexicon differs radically from everything that I have discussed here, both with respect to the basic theoretical background and to the particular formalization. None of the information Pustejovsky locates in lexical entries is found in the lexicon assumed in LDSNL, and a range of information is not expressed in the system at all. This is partly because the GLT conception clashes with Fodorian concepts, a modified version of which is part of the Relevance theory stance on concepts which I have adopted here. Thus, in the conception assumed here, there are no inferential roles associated with lexical items, which are, as pointed out earlier, instructions for concept formation. Other than these instructions, the LDSNL lexicon does not encode (lexical) semantic information, while GLT postulates rich semantic information in the lexicon. Furthermore, GLT employs feature structures both for the representation of information within lexical items, as well as for the lexical inheritance structure. Lexical inheritance structure models lexical semantic relations as sets of features which are represented in a tree-like structure such that a more general term is higher in the tree, and all lower nodes inherit the features from the more general node. The role of features and feature unification in LDSNL has been discussed in Chapter 4, but it is worth noting that LDSNL does not employ features in relation to concepts, nor in effect for concept combination[128]. The overall theoretical approach in GLT is generative in the sense that the infinite set of word meanings is attempted to be modelled by a finite set of features which can be recursively combined, although this claim is never fully spelled out. Theoretical differences notwithstanding, there is a sense in which GLT does provide an alternative to e*, which involves however some preliminary adjustments. Since words as such do not mean anything in the RT/LDSNL conception, lexical entries in GLT have to be re-interpreted as conceptual senses in order to compare the two approaches. Under this interpretation, what GLT is modelling is in effect part of the process of concept formation. After all, both GLT and RT/e* share the assumption that *open* means different things according to

128 In the sense that, as pointed out above, semantic combination is restricted to function-application, which can of course be represented as feature unification.

context. The question is then whether the formalization proposed by GLT can be employed for the more formal characterization of at least part of the process of concept formation which was left unformalized in the last chapter. One immediate restriction in this move is that, while GLT might help to formalize concept formation, it does not, as it stands, help with e* – there is no treatment of adjuncts, nor of ditransitive verbs. The only examples discussed are transitive verbs which change their meaning with respect to their object. The notion of shadow and default arguments is not discussed in sufficient detail to develop an account of e* out of it. However, even given these restrictions, there are problems with GLT.

2.3. Concept Formation as Feature Unification: Generative Concepts

The problem with taking GLT as a basis for conceptual enrichment is somewhat the inverse of the problem of taking it as a theory of lexical semantics – while the entries are too rich for the lexicon, they are too poor for concepts. Since the GLT entries, though rich, are restricted to information which is 'grammatically relevant', only some contextual differences in word meaning are expressed. Thus for example, since the difference between the change of state and the create senses of *bake* is partly tied to the mass/count distinction of the (D-argument of the) object, there is no difference between (10) and (11):

(10) John was baking a cake.

(11) John was baking a bread.

While the concepts involved here are probably similar, there still is the possibility to construe them differently on occasion; for example, the baking of bread might involve more preparation and time, different ingredients, different tools, etc. than the baking of cakes[129]. Yet this difference is not expressed in GLT, which in fact predicts that the two senses of bake are completely identical.

On the other hand, since the creative reading results from the unification of qualia values, in particular only from those where the object's qualia value explicitly states that the object is created by baking, all other cases fall under the

129 A nice way to make bread is to use sour dough baked on low heat in a wood fired stone oven. This doesn't work for cakes.

change of state reading, which is provided in the verb's entry. Thus, the meaning of (12) and (13) come out as the same:

(12) John was baking a potato.

(13) John was baking a flower.

But, as in the examples above, there are contexts where the baking of potatoes might differ from the baking of flowers. Furthermore, the change of state reading should be available for objects which are made by baking:

(14) John was baking a pizza.

In a context where the pizza is deep-frozen, the *bake* has no creative meaning, but given the way qualia structures unify, this reading is not obtainable, unless an ambiguity between pizzas and deep-frozen pizzas is postulated, which is what GLT is trying to avoid.

All the examples discussed so far point to problems with GLT which the theory according to its own aims should be able to handle. More general problems of this approach, as for example discussed in Fodor & Lepore (1998), include that GLT specifies just an arbitrary subset of world knowledge we have about things in the world, rather than about words. In principle, according to this criticism, there is nothing to be gained from writing the fact that cakes are made by baking into the lexicon. Another criticism, raised in Hunter & Marten (1999), concerns the use of typed feature structures for the representation of this kind of knowledge; feature unification as used in GLT does not support any logical reasoning, all lexical items have to be fully specified and be assigned a place in the feature hierarchy. Furthermore, context sensitivity has to be specified in advance as the presence or absence of particular features. In this sense, the system is completely unsuitable for modelling concept formation in the sense described in the last chapter.

The lack of context sensitivity and inferencing in GLT has been addressed by Lascarides, Copestake & Briscoe (1996) and Lascarides & Copestake (1998). They provide a conditional logic which models commonsense entailment and which interacts with the senses provided by GLT[130]. Under this view, the novel senses of GLT, derived by feature unification, are default predicates which are part of the meaning of the sentence in which they occur. The second step of interpretation takes these typed feature structures as input and

130 This work is discussed more extensively in Hunter & Marten (1999) and Wilson (1999).

combines them into a model of discourse processing in which notions of discourse coherence (e.g. 'elaboration', 'contrast') are defined. The combination of GLT with discourse processing is exploited to provide an interpretation of default senses. A default sense can be checked against the world knowledge base. For the pizza example in (14) above, for example, the default information that John created the pizza results from the lexical information that pizzas are created by baking. The world knowledge base then might specify that baking deep frozen pizzas implies merely a state of change. This information is more specific (pertaining to more specific pizzas) than the information about pizzas in general. Now the conditional logic specifies that more specific information overrides more general information, and thus the default interpretation does not survive under embedding into the discourse, and *bake* is (again) interpreted in its change of state sense. In the absence of contradicting or more specific information, lexical defaults are taken over into the discourse. The model thus provides a means to combine the lexical senses from GLT with world knowledge and furthermore offers some indication of how the two interact.

There are, however, still problems with this enriched version of GLT. First, the model inherits the problems of GLT noted above, namely that the different senses are not fine grained enough to distinguish different occasion specific concepts; different concepts with identical features (such as examples (10) - (13) above) are not distinguished at the discourse level, at least not if the defaults survive[131]. Secondly, this conception implies a rather unintuitive division of labour between lexical features and default reasoning; in the pizza example, the information that pizza baking (usually) means creating the pizza is lexical, while the information that baking a deep frozen pizza (usually) means changing the state of the pizza is part of the world knowledge, but the kind of information conveyed by the two statements is intuitively rather similar. A more general problematic point is the claim that default reasoning together with principles of discourse coherence replaces the notions of relevance and cognitive accessibility, so that the amended GLT model is taken to provide a genuine alternative to psychological or pragmatic models of natural language interpretation (cf. Lascarides & Copestake 1998, Wilson 1999), a point which I briefly discuss in the next section[132].

---

131 If the default interpretations are overridden, more information might be included, depending on the world knowledge base. In order to model enrichment in the sense assumed here, defaults would always be overriden and thus be pointless.

132 There are also a number of potential problems associated with the use of conditional logic with defaults for this kind of reasoning, which might be better formalized by employing default logic; cf. Hunter & Marten (1999: 30/31) for further discussion.

In conclusion, it appears that GLT, even when adjusted and amended, is not very suitable to model the process of concept formation relevant for the interpretation of e*. The approach suffers, from the perspective adopted here, from a fundamental misconception of the notions of lexical and world knowledge which results in counter-intuitive analyses and a general lack of clarity about what precisely is being claimed. In the next section, I discuss how these problems can be overcome.

3. Reasoning with World Knowledge

In this section I discuss work reported in Hunter & Marten (1999). Before doing so, however, I briefly clarify what this work is trying to do. As noted above, Lascarides & Copestake (1998) view their work as an alternative to purely pragmatic explanations. In contrast, the work discussed here takes a slightly different angle – a (potentially) computationally implementable statement of a theoretical explanation is an application of the latter to another domain (computers). In that sense it is not intended to provide an alternative, but rather a translation. For example, the substitution of the notion of accessibility by the notion of default is justified if, in the domain of computers, default corrresponds more or less to what accessibility is in the domain of the mind. Hence the point here is to formalize the analysis of the last chapter with a view to implementing it on a computer[133]. However, this is a preliminary discussion and a number of disclaimers apply. In the work reported here, Hunter & Marten assume that reasoning applies to the output of the parser, that is, an account is given of how world knowledge interacts with predicates of varying arity, but not of how these predicates are formed. However, the dynamics of this process could possibly be added into the picture at a later stage. Furthermore, there is no prioritization of assumptions, that is, the notion of relevance remains unanalysed in the version discussed here. The work thus is mainly concerned with modelling reasoning with world knowledge and e*-like predicate argument structures under the assumption that lexical items address concepts. Consequently, all reasoning with linguistic structures is located in the world knowledge, which means in particular that it is open to logical inferencing and does not run over typed-feature structures. As a further contrast to GLT, Hunter & Marten represent world knowledge as statements in default logic (as opposed to conditional logic) which incorporates classical first-order logic and provides an expressive and clear format for stating context

133 There might of course be a 'back-flow' from applied linguistics to theoretical linguistics.

sensitive reasoning[134]. I first introduce the system of Hunter & Marten (1999) and discuss it with respect to the theoretically motivated analysis of concept formation in the subsequent section.

3.1. A Default Logic Approach for Natural Language and World Knowledge

In the system described in Hunter & Marten (1999), the output of the parser is represented as function symbols which are translated into the logical language with recourse to world knowledge so that information from the parser and contextual default reasoning interact. Semantic information is represented as 'concepts', where a concept is a term of the logical language and accesses a set of logical statements (i.e. 'assumptions'), although the actual set accessed is at present left unspecified. Both these representations are part of the logical language (the world knowledge) expressed by default rules, so that the term accessing the set of assumptions may be part of assumptions in the set[135]. Reasoning with the output of the parser and world knowledge is characterized as comprising three related activities; commonsense checking, commonsense inferencing, and commonsense explaining. I introduce the formal architecture of Hunter & Marten with reference to these activities in what follows[136].

Commonsense checking establishes whether a given output from the parser is consistent with word knowledge facts. The notions of normal and unusual information can be checked with reference-to a (arbitrary) subset of default rules which might be called Normality default rules. For example, the idea that buttering toasts is usually done with a knife can be expressed with the following default rule:

(15) $$\frac{\text{Butter(X, the\_toast) : Butter(X, the\_toast, with(a\_knife))}}{\text{Butter(X, the\_toast, with(a\_knife)}}$$

The rule in (15) is a default rule. It states that, given Butter(X, the_toast) (i.e. the expression to the left of the colon, the 'precondition'), and furthermore given that Butter(X, the_toast, with(a_knife)) (i.e. the expression to the right of the colon, the 'jusitfication') is consistent with world knowledge, then Butter(X, the_toast, with(a_knife)) (that is now the expression under the line,

134 Cf. Reiter (1980) for a description of default logic.
135 Cf. also Rips (1995) for a similar conception of psychological concepts.
136 All examples in this section are from Hunter & Marten (1999), to which the reader is also referred for formal definitions of the notions introduced here and further discussion.

the 'consequent') can be inferred[137]. The rule is stated over expressions of the logical language, not over the output of the parser directly. It is thus a piece of world knowledge, expressing an aspect of what (most) people know about buttering toasts. With reference to this rule, the sentence in (16) can be checked, which Hunter & Marten assume is represented as (17) after parsing:

(16) John buttered the toast with a knife.

(17) butter(John, the_toast, with(a_knife))

The first step in the interpretation of (17) is to translate the 'lexical' function symbol butter (lower case) in (17) into the world knowledge predicate Butter (upper case), which is formally achieved by the world knowledge predicate Holds applying to the output of the parse:

(18)
$$\frac{\text{Holds(butter(X, Y, Z)) : Butter(X, Y, Z)}}{\text{Butter(X, Y, Z)}}$$

The Translation default rule in (18) effectively states that the lexical predicate *butter* can be interpreted as the conceptual predicate Butter if it is consistent with the world knowledge to do so. Furthermore, commonsense inferences can be stated over the output from parsing using Holds. For example:

(19)
$$\frac{\text{Holds(butter(X, Y, Z)) \& PartOf(X, X') \& PartOf(Y, Y') \& Human(X') \& Food(Y') : Holds(butter(X, Y)}}{\text{Holds(butter(X, Y))}}$$

The Facilitation rule in (19) allows the inference from the three place function symbol *butter* to the two place function symbol *butter* under the assumption that subject and object are human and food respectively, and that the inference is consistent with world knowledge. Holds then translates the inferred expression into a world predicate:

(20)
$$\frac{\text{Holds(butter(X, Y)) : Butter(X, Y)}}{\text{Butter(X, Y)}}$$

---

137 Technically, the inference is valid if no inconsistencies result with respect to the inferences derived, not with the total set of assumptions. It is presently left open which assumptions constitute relevant world knowledge.

Now the inferred expression in (20) can be used as a precondition of the Normality rule in (15). The information in (17) is thus not only consistent with the world knowledge, but can also be seen as redundant in the sense that the same information can be inferred by Normality rules.

Before proceeding, I briefly point out what this system of reasoning representation does. The model introduced by Hunter & Marten assumes that all semantic information is located in the world knowledge, which is represented as a (large) set of first-order logic statements and default rules. The output of the parser is taken to be uninterpreted. Interpretation is provided by taking this output, that is here predicate–argument structures of varying arity, and relating it to the predicate symbols which are in the world knowledge. Given the complete generality of the system, both enrichment (as in (15)) and inferences (as in (19)) can be stated. Although the system requires a large number of individual statements and rules, it has the advantage that steps of interpretation are explicitly and clearly statable. The following discussion includes further examples of how world knowledge interacts with natural language representations.

The example considered above involved information consistent with world knowledge. The following example, in contrast, is unusual:

(21) butter(John, the_toast, with(a_spade))

By using Translation rule (18), the lexical predicate is translated into a world predicate:

(22) Butter(John, the_toast, with(a_spade))

Furthermore, by the rules in (19) and (20), the proposition in (23) can be inferred from (21):

(23) Butter(John, the_toast)

The assumption in (23) can in turn be used with the Normality default rule in (15) to give (24):

(24) Butter(John, the_toast, with(a_knife))

Under the assumption that the world knowledge includes the information that (22) and (24) are contradictory, the information in (21) can be flagged as unusual with respect to world knowledge. Note that the system merely indicates why (21) is unusual, but does not imply any resolution, which has to be stated separately. This contrasts with the GLT position where principles regulate the interpretation of defaults, e.g. in this example, the default inference (if it was arrived at) would be suppressed given that *with a spade* is part of the sentence. The aim here is more modest, since inconsistency is checked, but not resolved.

Commonsense explaining involves deriving new predicates, either world knowledge predicates or Holds inferences, so that commonsense inferencing as in (19) above can be seen as an instance of commonsense explaining. Another instance of commonsense explaining involves term substitution. By using term substitution, function symbols can be replaced by other, possibly more complex, function symbols, which might be more meaningful in a given context. That is to say, substitution licenses the translation of one predicate into another, or possibly several other predicates if doing so is consistent with world knowledge. For example, for the n-ary function symbol flies(α_1, ..., α_n), where α_1, ..., α_n are terms, the following substitutions might be useful:

(25a) Sub(flies($\alpha_1,\ldots,\alpha_n$), moves(through_the_air($\alpha_1,\ldots,\alpha_n$)))

(25b) Sub(flies($\alpha_1,\ldots,\alpha_n$), moves(through_a_trajectory($\alpha_1,\ldots,\alpha_n$)))

(25c) Sub(flies($\alpha_1,\ldots,\alpha_n$), moves(quickly($\alpha_1,\ldots,\alpha_n$)))

(25d) Sub(flies($\alpha_1,\ldots,\alpha_n$), moves(swiftly($\alpha_1,\ldots,\alpha_n$)))

Which one of possible substitution rules is appropriate is of course context dependent. Consider the examples in (26) and (27):

(26) flies(this_helicopter, to(the_island))

(27) flies(time, like(an_arrow))

These two examples can be rewritten using substitution with the following default rules (where T stands for zero or more further term variables):

(28)
$$\frac{\text{PhysicalObject}(X) : \text{Sub}(\text{flies}(X, T), \text{moves}(\text{through\_the\_air}(X, T)))}{\text{Sub}(\text{flies}(X, T), \text{moves}(\text{through\_the\_air}(X, T)))}$$

(29)
$$\frac{\neg\text{Aircraft}(X) : \text{Sub}(\text{flies}(X, T), \text{moves}(\text{quickly}(X, T)))}{\text{Sub}(\text{flies}(X, T), \text{moves}(\text{quickly}(X, T)))}$$

Application of these rules (and assuming the relevant world knowledge facts, e.g. PhysicalObject(this_helicopter)) results in the commonsense explanation of (26) and (27) as (30) and (31):

(30) moves(through_the_air(this_helicopter, to(the_island)))

(31) moves(quickly(time, like(an_arrow)))

As these examples show, term substitution thus provides another means of using default inferences in reasoning with parsed natural language strings and world knowledge. It should be noted that there is considerable overlap between substitution and Holds inferences; the information provided by (30) and (31), for example, could equally have been arrived at by a default rule using the Holds predicate. This adds to the expressivity of the system, which can be constrained according to application.

To summarize, the system advocated in Hunter & Marten aims at providing a clear logic based formal account of how general reasoning interacts with output from parsing, in particular with predicate–argument structures of varying arity. The system is formulated in Default Logic, so that steps of inference are represented as default rules which license the commonsense checking, inferencing, and explaining of logical input structures, while being able to express context sensitivity and uncertainty. The underlying assumption is that natural language expressions address concepts, that is expressions of the logic, and that all interpretation is inferential, expressible in the system. The approach thus overcomes the problem of postulating two distinct levels of interpretation, such as feature structures and conditional logic in GLT, a division which is poorly motivated. The system is furthermore very expressive, since it offers several ways to derive particular information and can as such be suitably restricted given the need of particular applications.

3.2. Discussion

The system of Hunter & Marten outlined in the preceding section does not provide a formal analysis of concept formation. However, there are a number of traits which make it an attractive starting point for the development of such an analysis. As the examples discussed above show, the system handles examples similar to the enrichment cases discussed in the preceding chapter. For example, the enrichment of a constituent, that is the inference from (32) to (33) is expressible:

(32) John buttered the toast.

(33) John buttered the toast with a knife.

Similarly, the inference from (34) to (35):

(34) John buttered the toast with a fork.

(35) John buttered the toast.

Furthemore, by substitution, (36) can be translated to (37):

(36) John buttered the toast.

(37) John applied an even layer of butter to the toast.

These three inferences are stated within one system of knowledge representation, which is capable of formalizing context sensitive and uncertain reasoning. By using explicit facilitation and translation rules, inferences can be incrementally added or retracted according to context. Furthermore, prototypical information can be represented by individually specified normality defaults. The expressive power thus exceeds typed feature structures and lattice theoretic representations, and provides a better means to represent processes of general reasoning. Even without a full prioritization of rules, I believe an at least approximate formalization of the interpretation of e^* can be achieved with this system.

In order to employ this system for providing an LDSNL interface to general reasoning, a number of assumptions have to be modified. Thus, in order to model the contextual aspects in the establishment of argument structure, the default rules have to interact with LDSNL trees during the parse, not after the parse is established. This extension is, given the overall architecture, feasible; note that default rules can be used with predicate and term variables, and that inferences from, for example, binary to ternary predicates are possible. This modification, in conjunction with suitably defined normality statements, could be used, at least in principle, to model what counts as extra inferential effects, which are required by Swahili applied verbs. However, the first task is to implement a notion of prototypical attributes for verbs (i.e. their lexical 'denotation') together with a set of, necessarily artificial, relevant contextual default rules. One particular quality of the system which might be employed in modelling reasoning with e^* is the relation between substitution and Holds, noted above. Substitution licenses the manipulation of predicate symbols; in a given context, one predicate can be replaced by another.

Hold, on the other hand is employed for the derivation of inferences from predicates. As noted above, there is overlap between substitution and Holds, which, if appropriately implemented, could be used to model the relation between concept strengthening and contextual effects. Under this analysis, substitution of a term is possible if it results, in conjunction with world knowledge, in the increase in Hold inferences. However, the first step for using the system is to build a conceptual knowledge base, which specifies a set of assumptions and rules associated with a given predicate, and how this set depends on contextual information, including information provided by other predicates in the tree.

In summary, then, the system described by Hunter & Marten (1999) models reasoning with predicates of varying arities, as implied by the e* analysis developed here, and in addition provides a format for analysing the role of world knowledge in interpretation. Thus while the default logic model does not provide an analysis of the interpretation of e*, it does, I believe, provide the necessary basic architecture.

4. Conclusion

In this chapter I have discussed aspects of the formalization of the interpretation of e*. While no such account has been developed, I have contrasted the typed feature structure based approach taken by GLT with the logic based approach taken by Hunter & Marten (1999), in particular with respect to the question of whether either approach could serve as a basis to model pragmatic concept formation as employed in the analysis of e*. The conclusion of this chapter is that, given a sufficiently expressive characterization of world knowledge, such an analysis might be feasible.

Chapter 8

Conclusion

1. Introduction

In this chapter I present a summary of the thesis and offer some concluding remarks.

2. Summary

In this thesis I have argued that subcategorization information provided by verbs is underspecified. Verbs may specify how many Ty(e) expressions, which I have argued include NPs and PPs, they minimally require in order to derive an expression of Ty(t), a proposition, but they are in general flexible with respect to how many Ty(e) expressions can optionally be introduced into the verb phrase. This view is supported by linguistic evidence which shows that arguments and adjuncts behave alike in a number of respects, so that in these cases the only difference between arguments and adjuncts is the obligatoriness of the former and the optionality of the latter. I have argued that an analysis of VP adjunction which treats the adjunct as a functor which takes the verb or the VP as argument does not adequately explain this fact, and that furthermore such a putative analysis is problematic from the point of view incrementality, which requires that unfolding tree structure is built in a step-by-step fashion and that no tree structure once established can be undone. It was shown that an Adjunction analysis implies the restructuring of trees, so that it is not a possible analysis within the overall LDSNL model. I have provided an alternative solution, which introduces an underspecified type value into the LDSNL system:

(1) *Definition of (e* –> t)*

$$(e^* \rightarrow t) \quad =_{def.} \{(t) \vee (e^* \rightarrow (e \rightarrow t))\}$$

The definition of the underspecified type value e* states that verbs with such a type can be used with any number of Ty(e) expressions, since the underspecified type can be recursively resolved by introducing an additional

Ty(e) expression. The introduction of the notion of underspecification into the type specification of verbs is motivated also by the fact that structural underspecification is employed in the LDSNL system for values of the tree -node and formula values so that the solution follows from the overall assumptions of the model. The definition of e* for verbs extends the notion of underspecification, which is used in the LDSNL system mainly for the analysis of preposed constituents, to the analysis of predicate–argument structure. In chapter 3, I have shown how this underspecified type can be employed for an analysis of VP adjunction which does not involve tree restructuring and which transparently expresses the relation between verbs and adjuncts as being parallel to the relation between verbs and arguments, since the verb can be seen to act as a functor for both arguments and adjuncts. The analysis has been described in detail, and I have shown how underspecified verbs contribute to the process of tree building, and how they interact with the LDSNL system. I have provided sample derivations for adjunction and adjunct extraction, and provided necessary lexical instructions. One of the consequences of this approach is that prepositions are analysed as functioning to build Ty(e) nodes, that is, as licensing the introduction of Ty(e) expressions into the tree, thus functioning like case marking in languages with a case system. I have provided the necessary lexical instructions for prepositions and have justified this view with respect to alternative analyses of prepositions.

After having provided the LDSNL formalization of underspecified verbs, I have turned to the question of the contribution of underspecified verbs in the process of utterance interpretation. I have discussed relevant work from the formal semantics literature and proposed a formulation of a rule of semantic interpretation for underspecified verbs, based on the analysis in McConnell-Ginet (1982), which models the incremental extensional interpretation of e*. However, this analysis assumes that model-theoretic interpretation is assigned directly to natural language expressions, a view which is argued in LDSNL to be wrong. I have presented this argument and shown that the semantic rules do not provide an adequate explanation of the role of underspecified verbs in utterance interpretation. In the context of this discussion I have introduced the Relevance theoretic notions of pragmatic enrichment and concept formation, which show that natural language expressions quite generally do not address fully specified concepts, that is expressions of the language of thought, but rather provide an instruction for the hearer to construct an occasion-specific ad-hoc concept which serves to derive particular occasion-specific inferential effects. I have argued that from this perspective the role of Ty(e) expressions in the verb phrase can be seen as an aid for the construction of the eventual concept addressed by the verb, so that verbal underspecification can be analysed as the overt syntactic reflex of the possibility to enrich concepts addressed by verbs under inclusion of information provided by optional Ty(e) expressions. The process of concept construction thus results in the establishment of the eventual meaning of the concept addressed by the verb, which includes the establishment of the eventual, occasion-specific arity of the predicate. This analysis shows how instructions from words, structural syntactic processes, and pragmatic inference interact in the process of utterance interpretation.

In Chapter 6, I have provided an analysis of applied verbs in Swahili, which have often been analysed as involving a process of valency changing, in that an additional Ty(e) expression is introduced. The argument developed in this thesis is that applied verbs instruct the hearer to built a stronger concept than a possible concept built from the base verb, so that additional inferential effects can be derived. I have argued that the instruction for concept formation may result in the introduction of an additional Ty(e) expression, but that this is not necessary. From this perspective, the syntactic facts follow from the underlying meaning of applied verbs, which is essentially pragmatic. I have presented evidence for this view and argued that it provides a better explanation of the facts than analyses which view applied verbs as only encoding a change in valency.

In the final chapter I have briefly discussed how the process of concept formation which plays a central role for the interpretation of underspecified verbs could be more formally stated. I have contrasted two approaches for representing conceptual and world knowledge and have argued that a logic based approach is more adequate than a typed feature structure based approach. I have then pointed out how a logic based approach to reasoning with natural language and world knowledge could be employed to provide an interface between the LDSNL system and a model of contextual reasoning, and how such a formalization might be used to model the conceptual aspects of the interpretation of underspecified verbs.

3. Concluding Remarks

The argument developed in this thesis is embedded in a model of utterance interpretation. Utterance interpretation involves different aspects of cognitive activity; the ability to relate a physical signal to lexical units, the ability to build structured representations from lexical instructions, and the ability to construct mental concepts from the information provided by words and contextual assumptions. The analysis developed in this thesis shows that these different abilities interact freely, and that, at least for the analysis of verb phrase interpretation, all aspects have to be taken into account. One result of this thesis is thus that the model of utterance interpretation outlined in the first chapter is a viable conception of the mental activity involved in assigning meaning to strings of words.

The thesis provides, in addition, an analysis of adjunction for LDSNL, an area of language structure which has not received much attention in the model previously. The ideas proposed here, that type values may be underspecified, and that verbs are unfixed until all Ty(e) expressions are introduced into the tree, may be developed further and may help to broaden the model's empirical coverage. Another problem which has been addressed in Chapter 3, and which has implications for the overall system, is the formal power of the rule Introduction, which I believe should be restricted in the manner outlined here to license only the building of subject nodes.

However, there are a number of unresolved issues with respect to the formalization proposed here. In particular, the solution for optional objects offered here, namely to postulate lexical ambiguity, is not optimal. Given the formulation of e*, it would be expected that optional NPs, in addition to PPs, are covered by the analysis in a more principled fashion. The problem with optional objects is that the analysis as currently formulated postulates that additional Ty(e) nodes are built by prepositions, a solution which is necessary to restrict the formal power of e* and to prevent rather large-scale overgeneration. Yet this aspect of the analysis prevents an elegant characterization of the phenomenon of optional objects. The relevant line of inquiry here, and also more generally is to investigate more thoroughly the structure of lexical entries, and to devise means to express the underspecified nature of verbs by lexical actions. More generally, it seems to me that the role of the lexicon in the LDSNL model currently is possibly too central. Although it is true that hearers build trees from lexical information, the current formalization threatens to be too powerful since lexical instructions can freely be added. I think that both restrictions on lexical actions, and the context sensitivity of lexical information are questions which should be addressed.

The thesis, then, provides a contribution to LDSNL and the study of utterance interpretation more generally. It explores the nature of underspecification inherent in natural language verbs and verb phrases and presents an argument for the view that natural language interpretation is a process in which phonological, syntactic, and pragmatic information is freely exploited in the overall goal to assign meaning to a wave of sound.

Bibliography

Alsina, Alex & Sam A. Mchombo, 1993, 'Object asymmetries and the Chichewa applicative construction', in Mchombo, ed., 17-45.

Ashton, Ethel O, 1947, *Swahili Grammar*, 2nd ed., Harlow: Longman.

Baker, Mark C., 1988, *Incorporation: A Theory of Grammatical Function Changing*, Chicago: University of Chicago Press.

Barwise, J. & R. Cooper, 1984, 'Generalized quantifiers and natural language', *Linguistics & Philosophy* 4: 159-219.

Barwise, Jon & John Perry, 1983, *Situations and Attitudes*, Cambridge, Mass.: MIT Press.

Bennett, David C., 1975, *Spatial and Temporal Uses of English Prepositions*, London: Longman.

Berwick, Robert & Amy Weinberg, 1984, *The Grammatical Basis of Linguistic Performance*, Cambridge, Mass.: MIT Press.

Blackburn, Patrik & Wilfried Meyer-Viol, 1994, 'Linguistics, logic and finite trees', *Bulletin of the Interest Group for Pure and Applied Logic* 2, 3-29.

Bouma, Gosse, Rob Malouf & Ivan A. Sag, 1997, 'Satisfying constraints on extraction and adjunction', draft ms., Groningen & Stanford Universities.

Bresnan, Joan, 1978, 'A realistic transformational grammar', in Moris Halle, Joan Bresnan & George Miller, eds., *Linguistic Theory and Psychological Reality*, Cambridge, Mass.: MIT Press, 1-59.

Bresnan, Joan, & Sam A. Mchombo, 1986, 'Grammatical and anaphoric agreement', *CLS* 22. 278-297.

Bresnan, Joan & Sam A. Mchombo, 1987, 'Topic, pronoun, and agreement in Chichewa', *Language* 63, 741-782.

Bresnan, Joan & Lioba Moshi, 1993, 'Object asymmetries in comparative Bantu syntax', in Mchombo, ed., 47-91.

Brody, Michael, 1995, *Logico-Lexical Form: A Radically Minimalist Theory*, Cambridge, Mass.: MIT Press.

Butt, Miriam & Wilhelm Geuder, eds., 1998, *The Projection of Arguments*, Stanford: CSLI.

Campe, Petra, 1994, *Case, Semantic Roles, and Grammatical Relations: A Comprehensive Bibliography*, Amsterdam/Philadelphia: Benjamins.

Cann, Ronnie 1993, *Formal Semantics: An introduction*, Cambridge: CUP

Carpenter, Robert, 1992, *The Logic of Typed Feature Structures*, Cambridge: CUP.

Carston, Robyn, 1988, 'Implicature, explicature and truth-theoretic semantics', in Ruth M. Kempson, ed., *Mental Representations,* Cambridge, CUP, 155-181.

Carston, Robyn, 1996, 'Enrichment and loosening: complementary processes in deriving the proposition expressed', *UCL Working Papers in Linguistics* 8, 61-88.

Charette, Monik, 1991, *Conditions on Phonological Government,* Cambridge: Cambridge University Press.

Chierchia, Gennaro, 1995, *Dynamics of Meaning,* Chicago: University of Chicago Press

Chomsky, Noam, 1957, *Syntactic Structures,* The Hague: Mouton.

Chomsky, Noam, 1964, 'Current Issues in Linguistic Theory', in Jerry A. Fodor & Jerrold J. Katz, eds., *The Structure of Language: Readings in the Philosophy of Language,* Englewood Cliffs: Prentice-Hall, 50-118

Chomsky, Noam, 1981, *Lectures on Government and Binding,* Dordrecht: Foris.

Chomsky, Noam, 1995, *The Minimalist Program,* Cambridge, Mass.: MIT Press.

Chung, Sandra, 1982, 'Unbounded dependencies in Chamorro grammar', *Linguistic Inquiry* 13, 39-77.

Chung, Sandra & Carol Georgopoulos, 1988, 'Agreement with gaps in Chamorro and Palauan', in Michael Barlow & Charles A. Ferguson, eds., *Agreement in Natural Language: Approaches, Theories, Descriptions,* Stanford, Cal.: CSLI.

Clements, George N., James McCloskey, Joan Maling, & Annie Zaenen, 1983, 'String vacuous rule application', *Linguistic Inquiry* 14, 1-17.

Cobb, Margaret, 1997, *Conditions on Nuclear Government in Phonology,* doctoral dissertation, SOAS, University of London.

Copestake, Ann, 1996, 'Underspecification and defaults in lexical semantic representation', Paper presented to the Max Planck Research Group, Berlin.

Copestake, Ann, Dan Flickinger & Ivan A. Sag, 1997, 'Minimal Recursion Semantics: an introduction', ms., CSLI, Stanford University.

Davidson, Andrew, 1999, *Linguistic (Ir-)Realities,* doctoral dissertation, SOAS, University of London.

Dowty, David R., 1979, *Word Meaning and Montague Grammar,* Dordrecht: Reidel.

Fillmore, Charles, 1988, 'The mechanisms of "Construction Grammar"', *BLS* 14, 35-55.

Fillmore, Charles, 1994, 'Under the circumstances (place, time, manner, etc.)', *BLS* 20, 158-172.

Fodor, Janet D., 1995, 'Comprehending Sentence Structure', in Lila R. Gleitman & Mark Liberman, eds., *An Invitation to Cognitive Science, vol. 1: Language,* 2nd edition, Cambrigde, Mass.: MIT, 209-246

Fodor, Jerry A., 1973, 'Three reasons for not deriving "kill" from "cause to die"', *Linguistic Inquiry* 1, 429-438

Fodor, Jerry A., 1975, *The Language of Thought,* New York: Crowell.

Fodor, Jerry A., 1981, *Representations,* Cambridge, Mass.: MIT Press.

Fodor, Jerry A. & Ernie Lepore, 1998, 'The emptiness of the lexicon: critical reflections on J. Pustejovsky's The Generative Lexicon', *Linguistic Inquiry* 25.

Fodor, Jerry A., 1998, *Concepts: Where Cognitive Science Went Wrong,* Oxford: Clarendon.

Frazier, Lynn & Janet D. Fodor, 1978, 'The sausage machine: a new two stage parsing model, *Cognition* 6, 291-325.

Gawron, Jean Mark, 1985, 'A parsimonious semantics for prepositions and CAUSE', *CLS* 21:2, 32-47.

Georgopoulos, Carol, 1985, 'Variables in Palauan syntax', *Natural Language and Linguistic Theory* 3, 59-94.

Gorrell, Paul, 1995, *Syntax and Parsing,* Cambridge: CUP.

Grice, Paul, 1967, *Logic and Conversation,* William James Lectures, reprinted in Grice, 1989, 1-143.

Grice, Paul, 1989, *Studies in the Way of Words,* Cambridge, Mass.: Harvard University Press.

Groenendijk, Jeroen & Martin Stokhof, 1991, 'Dynamic Predicate Logic', *Linguistics & Philosophy* 14, 39-100.

Heine, Bernd, Ulrike Claudi & Friedrike Hünnemeyer, 1991, *Grammaticalization: A Conceptual Framwork,* Chicago: University of Chicago Press.

Hinrichs, Erhard & Tsuneko Nakazawa, 1994, 'Linearizing AUXs in German verbal complexes', in John Nerbonne, Klaus Netter & Carl Pollard, eds., *German in Head-Driven Phrase Structure Grammar,* Stanford, Cal.: CSLI, 11-37.

Hopper, Paul J. & Elizabeth Closs Traugott, 1993, *Grammaticalization,* Cambridge: CUP.

Horrocks, Geoffrey, 1987, *Generative Grammar,* London: Longman.

Hukari, Thomas E. & Robert D. Levine, 1994, 'Adjunct extraction', in Erin Duncan, Donka Farkas & Philip Spaelti, eds., *Proceedings of the Twelfth West Coast Conference on Formal Linguistics*, Stanford, Cal.: CSLI, 283-298.

Hukari, Thomas E. & Robert D. Levine, 1995, 'Adjunct extraction', *Journal of Linguistics* 31, 195-226.

Hunter, Anthony & Lutz Marten, 1999, 'Reasoning with output from parsing using world knowledge', draft ms., UCL/SOAS.

Jackendoff, Ray, 1983, *Semantics and Cognition*, Cambridge, Mass.: MIT Press.

Jackendoff, Ray, 1998, *The Architecture of the Language Faculty*, Cambridge, Mass.: MIT Press.

Kamp, Hans & Uwe Reyle, 1993, *From Discourse to Logic*, Dordrecht: Kluwer.

Kay, Paul, 1995, 'Construction Grammar', in Jef Verscheuren, Jan-Ola Östman, and Jan Blommaert, eds., *Handbook of Pragmatics*, Amsterdam: Benjamins. Reprinted in Paul Kay, 1997, *Words and the Grammar of Context*, Stanford, Cal.: CSLI, 123-131.

Kaye, Jonathan, 1989, *Phonology: a Cognitive View*, Hillsdale, NJ: Erlbaum.

Kaye, Jonathan, 1995, 'Derivations and Interfaces', in Jacques Durand & Francis Katamba, eds., *Frontiers of Phonology: Atoms, Structures, Derivations*, Harlow: Longman, 289-332.

Kaye, Jonathan, 1997, 'Why this article is not about the acquisition of phonolgy', in Stefan Ploch & David Swinburne, eds., *SOAS Working Papers in Linguistics & Phonetics* 7, 209-220.

Kaye, Jonathan, Jean Lowenstamm & Jean-Roger Vergnaud, 1990, 'Constituent structure and government in phonology', *Phonology* 7.2, 193-231.

Kaye, Jonathan & Jean-Roger Vergnaud, 1990, 'Phonology, morphology and the lexicon', paper presented at the 1990 GLOW colloquium, St John's College, Cambridge University.

Kayne, Richard & Jean-Yves Pollock, 1978, 'Stylistic inversion, successive cyclicity, and Move NP', *Linguistic Inquiry* 9, 595-621.

Keenan, Edward L., 1996, 'The semantics of determiners', in Shalom Lappin, ed., *The Handbook of Contemporary Semantic Theory*, Oxford: Blackwell, 41-63.

Kempson, 1988a, 'The relation between language, mind, and reality', in Ruth M. Kempson, ed., 1988b, 3-25.

Kempson, Ruth M., ed., 1988b, *Mental Representations*, Cambridge: CUP.

Kempson, Ruth M., 1996, 'Semantics, pragmatics, and natural language interpretation' in Shalom Lappin, ed., *The Handbook of Contemporary Semantic Theory*, Oxford: Blackwell, 561-598.

Kempson, Ruth M., Malcolm Edwards & Wilfried Meyer-Viol, 1998, 'Resumptive pronouns in English and Arabic', in Stefan Ploch & Gary-John Scott, eds., *SOAS Working Papers in Linguistics and Phonetics* 8, 35-57.

Kempson, Ruth M., Wilfried Meyer-Viol & Dov Gabbay, 1999, *Dynamic Syntax,* (working title), ms., SOAS & King's College, University of London.

Kimenyi, Alexandre, 1980, *A Relational Grammar of Kinyarwanda,* Berkley, Cal.: University of California Press

König, Esther & Uwe Reyle, 1998, 'A general reasoning scheme for underspecified representations', ms., Universität Stuttgart.

Kula, Nancy Chongo, 1997, *H-Licensing and Vowel Harmony in Venda,* MA dissertation, SOAS, University of London.

Ladusaw, William A. & David R. Dowty, 1988, 'Toward a nongrammatical account of thematic roles', in Wendy Wilkins, ed., *Syntax and Semantics 21: Thematic Relations,* New York: Academic Press, 61-73.

Lakoff, George, 1971, 'On generative semantics', in Danny Steinberg & Leon Jakobovitz, eds., *Semantics,* Cambridge: CUP, 232-296.

Lambek, Joachim, 1958, 'The mathematics of sentence structure', *American Mathematical Monthly* 65, 154-170.

Lappin, Shalom & David E. Johnson, 1999, *Local Constraints vs. Economy,* Stanford, Cal.: CSLI.

Lascarides, Alex & Ann Copestake, 1998, 'Pragmatics and word meaning', *Journal of Semantics* 11, 41-65.

Lascarides, Alex, Ann Copestake & Ted Briscoe, 1996, 'Ambiguity and coherence', *Journal of Linguistics* 34, 387-414.

Lyons, John, 1977, *Semantics,* 2 vols., Cambridge: CUP.

McCloskey, James, 1979, *Transformational Syntax and Model Theoretic Semantics: A Case Study in Modern Irish,* Dordrecht: Reidel.

McConnell-Ginet, Sally, 1982, 'Adverbs and logical form: a linguistically realistic theory', *Language* 58, 144-184.

McGee Wood, Mary, 1993, *Categorial Grammars,* London: Routledge.

Marantz, Alec, 1993, 'Implications of Asymmetries in Double Object Constructions', in Mchombo, ed., 113-150.

Marten, Lutz, 1997, 'Licensing constraints for Swahili vowel harmony', in Stephen Markve, Evelien Pennings & Ayumi Tsukiashi, eds., *Plum 2: Proceedings of the Sixth Manchester Postgraduate Linguistics Conference,* 239-255.

Mchombo, Sam A., ed., 1993, *Theoretical Aspects of Bantu Grammar,* Stanford: CSLI.

Montague, Richard, 1973, 'The proper treatment of quantification in ordinary English', in J. Hintikka, J.M.E. Moravcsik & Suppes, eds., *Approaches to Natural Language*, Dordrecht: Reidel, reprinted in Richmond Thomason, ed., 1974, *Formal Philosophy: Selected Papers of Richard Montague*, New Haven: Yale University Press, 222-246.

Morrill, Glyn V., 1994, *Type Logical Grammar*, Dordrecht: Kluwer.

Nakamura, Masanori, 1997, 'Object Extraction in Bantu Applicatives: Some Implications for Minimalism', *Linguistic Inquiry* 28, 252-280.

Perrett, Denise, 1996, 'A comparison of tense construal in DRT and LDS', paper presented at SOAS, October 1996.

Perrett, Denise, fcmg., *Tense Construal in LDS*, doctoral dissertation, SOAS, University of London.

Pollard, Carl & Ivan A. Sag, 1994, *Head-Driven Phrase Structure Grammar*, Chicago: Chicago University Press.

Port, Robert F., 1981, 'The applied suffix in Swahili', *Studies in African Linguistics* 12, 71-82.

Pratt, Ian & Nissim Francez, 1997, 'Temporal Prepositions', Technical Report UMCS 97-4-2, Department of Computer Science, University of Manchester.

Pustejovsky, James & Branimir Boguraev, eds., 1996, *Lexical Semantics*, Oxford: Clarendon.

Pustejovsky, James, 1995, *The Generative Lexicon*, Cambridge, Mass.: MIT Press.

Ravin, Yael, 1990, *Lexical Semantics without Thematic Roles*, Oxford: Clarendon.

Reiter, R., 1980, 'Default logic', *Artificial Intelligence* 13, 81-132.

Reyle, Uwe, 1993, 'Dealing with ambiguities by underspecification: construction, representation, and deduction', *Journal of Semantics* 10

Rips, Lance J., 1995, 'The current status of research on concept combination'. *Mind & Language* 10, 72-104.

Sacleux, P. Ch., 1909, *Grammaire swahili*, Paris: Procure des PP. du Saint-Esprit

Sag, Ivan A. & Anna Szabolcsi, eds., 1992, *Lexical Matters*, Stanford, Cal.: CSLI.

Sag, Ivan A., 1998, 'Criteria for evaluating analyses of unbounded dependencies', Paper presented at the School of Oriental and African Studies, London.

Samsom, Ridder & Thilo C. Schadeberg, 1994, 'Kiinimacho cha mahali: kiambishi tamati cha mahali -ni', *Afikanistische Arbeitspapiere* 37, 127-138.

Schadeberg, Thilo C., 1992, *A Sketch of Swahili Morphology*, 3rd ed., Köln: Köppe.

Shibatani, Masayoshi, 1996, 'Applicatives and benefactives: a cognitive account', in Masayoshi Shibatani & Sandra A. Thompson, eds., *Grammatical Constructions: Their Form and Meaning*, Oxford: Clarendon, 157-194.

Sperber, Dan & Deirdre Wilson, 1986, *Relevance: Communication and Cognition*, Oxford: Blackwell.

Sperber, Dan & Deirdre Wilson, 1995, *Relevance: Communication and Cognition*, 2nd ed., Oxford: Blackwell.

Sperber, Dan & DeirdreWilson, 1997, 'The mapping between the mental and the public lexicon', *UCL Working Papers in Linguistics* 9, 107-125.

Swinburne, David, 1998, '"Hu" and the dynamics of Hebrew', paper presented at the 4th Conference on Afro-Asiatic Languages, London.

Swinburne, David, 1999, *<LINK> and the Dynamics of Utterance Interpretation*, doctoral dissertation, SOAS, University of London.

Tenny, Carol L., 1994, *Aspectual Roles and the Syntax–Semantics Interface*, Dordrecht: Kluwer.

Thomason, Richmond & Robert Stalnaker, 1973, 'A semantic theory of adverbs', *Linguistic Inquiry* 4, 195-220.

Tomaselli, Allessandra, 1997, 'Commentry on part 1: aspect, argument structure and case selection', in Ans van Kemenade & Nigel Vincent, eds., *Parameters of Morphosyntactic Change*, Cambridge: Cambridge University Press, 134-146.

Wald, Benji, 1993, 'Longterm evolution of the syntax of discourse and the Swahili person markers', in: Jaap van Marle (ed.), *Historical Linguistics 1991: Papers from the 10th ICHL, Amsterdam*, Amsterdam/Philadelphia: Benjamins, 325-341.

Wilson, Deirdre & Dan Sperber, 1999, 'Truthfulness and Relevance', draft ms., UCL, University of London.

Wilson, Deirdre, 1999, 'Review Lecture of Lascarides, Copestake & Briscoe 1996 and Lascarides & Copestake 1998', UCL Pragmatics Reading Group.

Zaenen, Annie, 1983, 'On syntactic binding', *Linguistic Inquiry* 14, 469-504.

Zwarts, Joost, 1997, 'Vectors as relative positions: A compositional semantics for modified PPs', *Journal of Semantics* 14, 57-86.