<LINK> and the Dynamics of Utterance Interpretation

David Thomas Swinburne

Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
University of London
April 1999

Department of Linguistics
School of Oriental and African Studies

Abstract

The <LINK> operation allows the derivation of inferential units as the result of syntactic processing. The same operation derives structure across a diffuse range of data; the differences between these are attributable to lexical information and the dynamics of the process.

The thesis examines a range of data where clauses depend on some element in the main clause for interpretation. These are assigned structure as <LINK> trees in the framework of Labelled Deductive Systems for Natural Language outlined in Kempson, Meyer-Viol & Gabbay, (*Dynamic Syntax: the Deductive Flow of Natural Language*, in prep.). I describe the approach to utterance interpretation this proposes: linguistic structure is built incrementally according to lexically encoded procedural instructions and rules of construction constrained by pragmatic principles. This allows a perspective where problems previously divided between syntax, semantics and pragmatics can be addressed in a more unified manner. I outline the <LINK> operation as it has been developed for relative clauses. This allows several trees to be built for the same utterance which are connected by a having a node description in common. I argue that this operation can be extended to cover the following data: extraposed relative clauses, reduced relative clauses, adjunct predicates and parenthetical constituents. To this end I introduce type (p) into the framework, which allows the representation of non-tensed propositions and I modify the <LINK> operation to allow the creation of trees from discontinuous input. I develop context-dependent lexical rules to capture the difference between modifying and predicative uses of lexical items. The interaction of processing tasks, compilation and lexical information determines the precise nature of the structures which are built. The update procedure I develop allows a uniform characterisation of the way structure is derived across different contexts, shedding new light on the dynamics of building interpretation.

Acknowledgments

I am very glad to have this opportunity to thank all the people who have helped, encouraged and supported me during the writing of this thesis.

This work was supported by a PhD Studentship from the Humanities Research Board of the British Academy, which I gratefully acknowledge.

Ruth Kempson has been a wonderful supervisor and a constant source of inspiration. I am profoundly grateful to her for all her support. Ruth has guided and encouraged me through every stage of this work; she has been there to discuss anything at any time, always ready with patience, insight, good humour and warmth. It has been an honour and a privilege to work with her.

Lutz Marten is a true friend and an inspiring colleague. He has been with me along the way on my journey in linguistics – it would not have been the same without him.

In my time in the Linguistics Department at SOAS, the people here, students and staff, have made it a great place to study. I would like to thank the many friends and colleagues who have made my time here such an enriching experience. In particular, I would like to thank Monik Charette, Stefan Ploch, Denise Perrett, Taeko Maeda, Caroline Fisher, Nancy Kula and Trevor Marchand. Thanks to Akiko Kurosawa for her understanding about the office.

Thanks are also due to Bruce Ingham, Shalom Lappin, Wynn Chao, David Bennett, Howard Gregory and Andrew Simpson, and to everyone else who has made Bloomsbury such a stimulating place to be.

Becoming involved in doing a PhD happens over time: Bob Chard and Ross King both encouraged me when I needed it.

A number of friends were able to convince me from their own experience that I could do it. Thanks to Melanie Green, Emma Stone and especially to Jo Grant who never stopped encouraging and cajoling me. Jo had to put up with more than her fair share of listening to my moans, and managed never to run out of reasons why, despite my protestations to the contrary, everything was, in fact, fine.

I have been fortunate to be able to count on the love and support of many friends in London who have all helped me in their different ways. Thank you to Alan Gutteridge, Anna Souhami, Aniel Biltoo, Nathalie Lemay Palmer, Young Soon Musafiri, Martin Kupar, Kate Mayberry, Emily Doncaster, Charles Richardson and Erica Whyman. Thanks, too, to Sonya Blizzard and Steve Ross for solace in the country.

Steve Cook deserves a special mention for all that he has done for me. I owe him a great deal.

Other friends, though more distant, have provided support from afar – the wonders of e-mail – and have made the occasional personal appearance. I would like to thank William Stewart, Alon Levkowitz, Michael Wills and Kiri Schultz, Andrew Dolbey and Lizanne Kaiser.

Keith Griffiths has made an enormous difference to my life over the last few months. I thank him for putting up with me when I have been far from my best, and for looking after me. I will forever be grateful for his love and support.

My family are a continual source of strength and support; being able to turn to them, and rely on them, has always been a great comfort. Thanks to Rachel and Natalie, Sally and Raf; Melanie and Steve, and Charlie and Angus, the latest arrivals.

Finally, I would like to thank my parents for everything they have done for me over the years. They have constantly supported me in what I wanted to do, only rarely pausing to ask why. They have been with me every step of the way, and their love and support has enabled me to reach where I am now. I dedicate this thesis to them with all my love.

Contents

Chapter Three

Extending <LINK>:

P-predication and Reduced Relative Clauses

Chapter Four

Discontinuous Constituency and Type (t) <LINK>:

Extraposed Relative Clauses

Chapter Five

Discontinuous Constituency and Type (p) <LINK>:

Extraposition and Adjunction

Chapter Six

Non-integrated Constituency and <LINK>:

Parenthetical <LINK> structures

Chapter One

Introduction:
Labelled Deductive Systems for Natural Language

1.0. Introduction

This thesis is a study of the <LINK> operation in Labelled Deductive Systems for Natural Language, a framework for modelling utterance interpretation by the incremental building of structure.[1] The <LINK> operation allows for the construction of multiple semantic trees for an utterance provided that some node description is held in common. I investigate how the <LINK> analysis can be developed to account for a range of adjunction structures.

I am concerned with the way that individual lexical items contribute to the process of structure building when used as adjuncts, and how this relates to their more general properties. I am also concerned with how the structure built depends on the dynamics of the process. I conclude that various adjunct structures can be analysed as <LINK> structures; lexical items map onto predication in a regular way, and these predicates are then conjoined with the main part of the sentence.

The significance for the framework is in terms of extending the empirical coverage and developing the theoretical machinery. Specifically I consider the formulation of the <LINK> operation, the application of transition rules and the way that movement of the pointer is effected.

The analysis provides further evidence of the advantages offered by the dynamic approach espoused in the LDSNL framework. A wide variety of phenomena, viewed as disparate problems, can be given a unified analysis; that is to say, application of the <LINK> operation will successfully derive structure

[1] This framework is developed in Kempson, Meyer Viol & Gabbay (in prep.).

in a number of environments; the differences are seen to stem solely from independent factors in the lexical input.

In more general linguistic terms, this gives a less stipulatory account of empty elements required to fulfil semantic requirements. The 'missing argument' is only supplied dynamically; so adjuncts do not require independent licensing, but depend on the felicity of the pragmatic effects achieved in any particular context. My concern is primarily to investigate the processes by which structure is built; but the objects derived by these processes should feed into a more general theory of interpretation.

In this chapter I introduce the basic assumptions of Labelled Deductive Systems for Natural Language, discussed in Kempson, Meyer Viol & Gabbay (in prep.), and outline how the rest of the thesis develops.

1.1. Underspecification and Utterance Interpretation

The starting point for the approach espoused by Labelled Deductive Systems for Natural Language (LDSNL) is the fact that linguistic items do not exhaustively specify the representation onto which they map - the linguistic system should not be characterised as an algorithm which will uniquely specify a representation for some input, given that the same input may be mapped onto a number of different structures. The aim of a linguistic theory, rather, is to explain how the possibilities arise: what is encoded in lexical items across contexts such that this information will combine with the general principles of the system to give the correct possibilities of interpretation.[2] Although the criteria for making choices is left to pragmatic considerations,[3] those choices are made on-line, rather than by constructing several structures and choosing between them as output at the end stage of some syntactic component. Underspecification is built into the system.

---

[2] See Kempson (1988).

[3] That is to say by means of the general constraints on inferential processes rather than specialised linguistic rules.

This is an extension into linguistics of developments in pragmatics and approaches to utterance interpretation. The context dependent nature of the interpretation of individual lexical items is noted by Bar-Hillel (1954). Grice (1975) is credited with introducing the idea that the derivation of interpretation for natural language involves an inferential stage, as well as a level of decoding. He seeks to explain how it is that what is communicated by a speaker goes beyond the literal meaning of their words, claiming that conversation is governed by generally accepted norms. Grice introduced the idea of implicature: an implicature is an assumption that has to be built into the interpretation of something that a speaker says in order to preserve the assumption that the Co-operative Principle and the associated maxims have been observed.

While it is generally agreed that the task of a pragmatic theory is to explain how it is that 'extra' meaning is built into an utterance according to the context of usage, the idea that pragmatics should determine aspects of the linguistic interpretation is due to Sperber and Wilson (1986/1995). In their development of Relevance Theory, they extended the domain of pragmatics to consider the role of inferential processes in the development of the propositional form of an utterance, (i.e. the truth conditional content), and the explicatures associated with it. Explicatures are a direct inferential development of what is encoded.[4] Sperber and Wilson model interpretation as a combination of decoding and inference, and seek cognitive explanation for the way that content is enriched in the process of utterance interpretation.[5] They say that human cognition is geared towards the deriving maximal relevance from any input processed. Relevance is defined by them in terms of achieving inferential effects, where new information interacts with existing contextual information. The drive to derive contextual effects is balanced by the desire to keep processing costs to a minimum. They introduce the concept of Optimal Relevance which strikes a balance between the two, and determines the way interpretations are developed. A hearer automatically assumes that a speaker intends to achieve adequate contextual

[4] Cf. Wilson and Sperber (1993), Carston (1998).
[5] Cf. Sperber and Wilson (1995).

effects using the most efficient stimulus at their disposal. All linguistic input is processed on this basis: the form of language will reflect this.

LDSNL takes the characterisation of utterance interpretation as a combination of decoding and inference and places this at the heart of the grammar. The defining features of natural language interpretation on this approach are the way in which linguistic input systematically underdetermines the representation ultimately derived for an utterance, and the way that structure is built up inferentially. Linguistic input has to be enriched subject to the general cognitive principles espoused in Relevance Theory. LDSNL is set up to model the role of inference in natural language interpretation and how this interacts with encoded meaning. This approach provides new insights into syntactic phenomena.

1.2. The LDSNL Perspective

The framework of Labelled Deductive Systems for Natural Language has been developed by Kempson and associates, and is presented in Kempson, Meyer Viol & Gabbay (in prep.). The present discussion is based on that work. I present the pertinent details of the formalism in chapter two.

LDSNL models utterance interpretation as an inferential task; a process whereby interpretation is built up incrementally on the basis of lexical input. That is to say, utterance interpretation is looked at from the perspective of the hearer: words provide the input to a reasoning task where the overall goal of the task is to derive a truth evaluable interpretation. At each stage, the structure building operation is directed by a (highly constrained) set of parsing principles and procedural information provided in the lexical entries of words. The primary focus of this approach is the process of structure building; syntactic restrictions are seen to fall out from the dynamics of arriving at a semantic representation. Derivations are represented on a step by step basis.

The framework develops a parsing schema where the goal is to derive a semantically interpretable formula. Different submodules of the system allow the manipulation of discrete types of information, while the whole combine to describe the step by step process by which structure is built up. The process is

driven by lexical information. This divides into labelling information (eg the type specification associated with a lexical item) and formula information (equivalent to the conceptual information contributed by a word).

The claim is that linguistic structure is defined over, rather than for, a string. The model builds annotated trees incrementally, reflecting each stage of the parse process. These correspond to propositional content: at each non-final stage, the tree is in some way incomplete. The parsing process, then, involves a sequence of partial tree descriptions, where each shift from word to word provides an incremental update. Underspecification is allowed both syntactically and semantically. An example of the latter is the case of pro-forms, which, although they do not uniquely determine the conceptual unit which must ultimately be instantiated in the representation, do provide the requisite instructions for the identification of such a unit. Syntactic underspecification is found in wh-elements which are identified only as holding somewhere below the root node of the tree (using the Kleene star operator), and where their position is identified according to the general requirements of the parsing process.

The way in which lexical information is conceived is essentially procedural - a series of instructions on how to build a representation. Thus, the labelling information indicates the way in which the parts combine, while the formula information indicates how to map on to some kind of denotation. The association of concepts with individual words is not addressed in the present work.

Fundamental to this approach is the claim that there is only a single level of linguistic representation, a level of logical form derived directly from the surface string without positing any intermediate levels of structure or movement operations. Nor is there any discrete mapping operation from some syntactic representation onto a discourse level, as modelled in Discourse Representation Theory, Kamp & Reyle (1993). Indeed, there is no one to one mapping between the input string and the eventual representation of the meaning. There is no possibility of this given the assumptions being made here. Each word will specify the possible strategies that the parser is licensed to do in terms of

structure building. In this way the system reflects the underdeterminacy of the information supplied to the linguistic system; pragmatic choices are made on-line. Correspondingly, it is not the case that the linguistic system derives multiple outputs.

What is of greatest interest in this framework is the *process* of structure building; what is eventually derived is a structure reflecting the content associated with a string of words, but it is how this structure is derived that is of primary concern. This is seen as incremental: partial trees map the stages of development. The dynamics of tree development will be addressed in chapter two.

The present approach changes the perspective on grammaticality.[6] Standardly it is assumed that a string is either well formed or not, according to whether or not the grammar licenses it as a licit structure. Note the current claim, however, that syntactic restrictions are a consequence of the dynamics of the structure building operation: the notions of grammar and parser are here being conflated. It is not that the parser performs operations to assign structure to a string and the grammar then determines whether or not such structure is acceptable in language 'X'. What determines the well formedness of a string is whether or not the parser can assign structure to it. There are cases where the derivation simply 'crashes'; for example, because there are insufficient arguments to fulfil the combinatorial requirements of a predicate. There may be other cases where the degree of processing effort will affect judgments about the acceptability of an utterance. Finally, the apparent acceptability may depend on the pragmatic effects associated with a particular processing strategy. If these do not accord with the conceptual content, the example may seem illicit.

1.3. Labelled Deduction and Natural Language

Labelled Deductive Systems (Gabbay (1996)) provide a means of describing inference systems where additional information about each stage can be

[6] Strictly speaking, overall well-formedness for an utterance can only be judged in a particular context on a certain interpretation.

represented explicitly, giving a detailed description of the way in which the whole is established. This allows direct representation of the information which controls the combination of premises and how this combination proceeds: discrete sub-languages are defined which represent different aspects of the system.

For natural language such an approach provides the means to integrate directly procedural information and conceptual information, and to display the way that the lexicon may encode both procedural and conceptual information. Procedural information is information on how to build trees, whereas conceptual information is the denotational inforamtion which a word contributes to a representation.

The declarative unit represents the basic building block of the system and is made up of two units of information, the type and the formula. This is represented as in (1).

(1) Formula (john'), Type (e)

As will be seen in the next chapter, these constitute information which is annotated on tree nodes, and are derived from the process of utterance interpretation.

1.3.1. Type

This is the logical type, (Ty), projected by words, describing what they are and constraining their combinatorial possibilities, and is similar to the way in which subcategorisation is represented as requirements in Categorial Grammar: e.g. (e), (t), (e→t).

Thus, an entity (e) has no further combinatorial requirements; this provides an argument (and is roughly equivalent to a DP), whereas a one place predicate such as an intransitive verb would be of type (e→t) requiring an entity in order to give a truth value. The steps of modus ponens map onto function application over the formula, according to the Curry-Howard isomorphism, as shown in (2). The type specification, then, provides a label to the formula.

(2)

$$\boxed{\begin{array}{l} \alpha : P \\ \beta : P \rightarrow Q \\ \beta(\alpha) : Q \\ \gamma : Q \rightarrow R \\ \gamma(\beta(\alpha)) : R \end{array}}$$

The system of Labelled Deductive Systems for Natural Language developed in Kempson, Meyer Viol & Gabbay (in prep.) adds the type (cn), common noun, to represent the internal structure of noun phrases.

A noun brings to the process a variable, itself of type (e) and a predicate of type (e→cn), the whole giving a unit of type (cn).

(3) man (cn) = U (e) man (e→cn)

This variable may then be bound by determiners, which are assigned the specification type (cn→e). Note however that this involves an operation of term-variable binding and not of function application (see 2.3.2.2.).

In chapter three I introduce the type (p). This is used to distinguish between different types of predication; a predicate which maps onto (t) is fully tensed and hence truth evaluable. Predicates which map onto type (p) are not truth evaluable in that they are not specified for temporal location; while intuitively they do describe some state of affairs, it is not anchored in a time flow. In order to be interpreted they must contract some temporal label.

1.3.2. Formula

The formula, (Fo), represents the conceptual content associated with words: individual constants, predicates, variables, place holding meta-variables.

This propositional content is defined in terms of a lambda calculus.

Words map onto a particular concept, and a combinatorial type associated with that:

(4)	Noun	Fo(melanie'), Ty(e)
	Verb	Fo(laugh'), Ty(e→t)
	Adjective	Fo(fabulous'), Ty(e→t)

The way in which premises combine is illustrated below for a transitive verb. What is derived from the combination of the words is a propositional formula with label (t), which represents the overall meaning.

(5) John kisses Bill.

(6) Fo(john'), Ty(e)
Fo(kiss'), Ty(e→(e→t))
Fo(bill'), Ty(e)
Fo(kiss'(bill')), Ty(e→t)
Fo(kiss'(bill')(john')), Ty(t)

Note that in the above, agreement information is not represented, nor is tense. These provide separate labelling information which is not relevant to the present discussion. Such information will be introduced as it arises in the analyses in the rest of the work.

While this provides a schematic representation of how meaning comes together, the actual process is more complicated. It is described in a more detailed step by step way using semantic trees, as will be discussed in the next chapter.

1.3.3. Tasks

The overall goal of linguistic processing is to derive a proposition of type (t). This can be broken down into sub-requirements, into the requirements for a subject, type (e), and a predicate, type (e→t), which can combine to give a proposition.

These constitute the basic requirements of the system. Lexical input is needed which will fulfil these requirements.

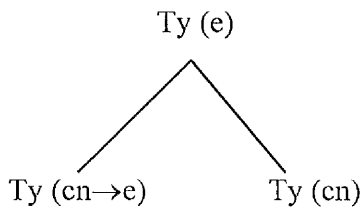
The process of fulfilling a requirement is called a task. Tasks are characterised in terms of types, hence e-task, t-task. A task may be made of sub-parts, as discussed for the t-task. Other examples are given in (7) and (8).

(7) e-task: Ty(cn→e), Ty(cn)

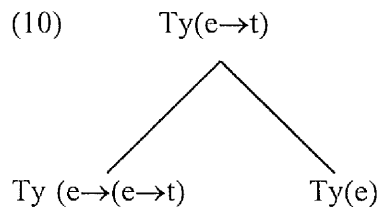
(8) (e→t)-task: Ty(e→(e→t), Ty(e)

These can be represented graphically as pairs of nodes, as shown in (9) and (10) respectively.

(9)

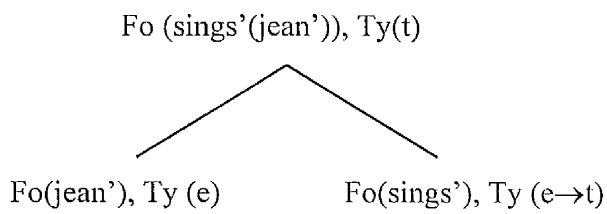


(10)



Formula information can also be appended to these node descriptions, as shown in (11).

(11)



I return to this in chapter two.

1.4. Outline of the Thesis

The idea of <LINK> is very simple. The basic claim is that the information which a word supplies in the process of building a tree structure can also be used in the process of building a new tree structure. The restrictions on this are that the new tree task be started from the existing one, and that a <LINK> relation is contracted between them. This is the means used to build structure for

relative clauses in the framework. The hypothesis I investigate is that this operation should be able to account for a wider range of data which display some degree of dependency between two clauses, and where one clause lacks an element. The way the rest of the thesis is organised is outlined below.

In chapter two I introduce the formal mechanisms by which structure is built in Labelled Deductive Systems for Natural Language, based on the system outlined in Kempson, Meyer Viol & Gabbay (in prep.). The system builds semantic trees where the nodes are annotated with declarative unit information indicating the type and formula information holding at that node. The location of the tree nodes can be explicitly described, and underspecification of the location is achieved using the <*> operator. The initial goal of the process is to derive a proposition of type (t). The process of tree development is effected by transition rules which allow for the construction of nodes and movement between nodes. The pointer marks where in the tree the structure building process is at any stage in the derivation. The transition rules are utilised in lexical entries which indicate the update function performed on the tree as the result of lexical input; these can only apply in the correct triggering environment. I give sample lexical entries for English, and illustrate how the process of structure building operates. I then introduce the <LINK> operation proposed in Kempson, Meyer Viol & Gabbay (in prep.) to derive structure for relative clauses. This allows for a new tree to be built, which has to share a node description with the original tree. I indicate how this operates for restrictive and non-restrictive relative clauses. All of this sets up the machinery which is used in the rest of the thesis.

In chapter three I extend the <LINK> analysis to account for reduced relatives. These are formed by adjective phrases, prepositional phrases, noun phrases, verb participle phrases and infinitives. I note that all of these categories require the support of the copula to constitute well-formed main clauses; but that this is absent in the reduced relative. I examine the contribution of the copula to building structure in LDSNL, in terms of its syntactic and semantic properties, and conclude that its function is only to supply tense. This leads me to consider predication without tense; I introduce a new type into the system,

type (p), which indicates a proposition with no temporal location. The categories given above will be defined in terms of this type; I term these the basic, or p-predicates. I discuss the way in which predication and modification affect tree structure, and how this is to be encoded in lexical entries. Lexical entries are context dependent; therefore specific actions are associated with specific triggering environments in the tree configuration, and I give examples of how this works. I define a <LINK> rule for type (p) which builds structure for the reduced relatives in the same way as for full relatives, and creates L-trees of type (p). The requirement that the shared node has to be the subject of the relative falls out from the general way that the subject node is built in English. I show how structure building develops for the reduced relatives.

In both of the above cases where the <LINK> operation is invoked, it straightforwardly maps onto a new tree from a node in the existing tree, where that node provides the node description to be shared between the two trees. I next examine cases where the <LINK> operation is not launched from the node which is identified as the shared one. This necessitates revision of the <LINK> rule.

Chapter four examines the extraposition of tensed relative clauses. I outline the basic properties of extraposed relatives and review the literature. Putative restrictions on the sort of noun phrase and the sort of predicate that can be used in this construction can be overcome in context; the restrictions are related to function. There is general dispute over the point of attachment of the extraposed relative: should the whole noun phrase or just some subpart of it be antecedent to the relative? I argue that the reason for this tension is that, in fact, either can be the case for extraposed relatives. I define the terms 'restrictive' and 'non-restrictive' specifically in terms of the <LINK> operation. A restrictive <LINK> tree has the metavariable supplied by the common noun as the shared node; a non-restrictive <LINK> tree has the whole of the e-node description as the shared node. In the case of extraposed relatives either structure may be available. This depends on the way that the e-node has been processed. If compilation takes place in the e-task, it is sealed off, and the L-tree must be non-restrictive. If however, only completion occurs then a restrictive L-tree can be

built. I relate this to the way that determiners function. The facts of extraposition are accounted for by redefining the <LINK> to allow for discrepancy between the launch location and the shared node; the relativiser acts as a pronoun instantiating a search procedure.

In chapter five I examine the extraposed form of the p-predicate reduced relatives. These have been analysed in the literature as instances of secondary predication. I distinguish between predicates which are subcategorised for and which lie outside the remit of the present work, and adjunct predicates. The <LINK> operation gives exactly the right sort of structural analysis for these without recourse to special mechanisms. Thus, the same mechanism for deriving structure is used as in the case with continuous input, deriving L-trees of type (p). Citing evidence from quantification, I argue that these are never restrictive. I relate this property to the accessibility of potential antecedents. Unlike the tensed relatives, in this case there is no lexically triggered search procedure; the shared node has to be identified from the child nodes of the pointer location. I then consider the interpretation of these L-trees. The sorts of connection that can be inferred between the two predicates are determined by the interaction of pragmatics and the structure. In this case the structure is two conjoined propositions sharing a temporal index, and thereby marked as simultaneous.

Chapter six is concerned with L-trees which are not integrated into the t-task of the main tree, the parentheticals. As in other cases of the p-predicates, some predication structure has to be built, and this can be achieved by the <LINK> operation. Parentheticals illustrate the advantages of the present approach; the lack of syntactic interaction with the main sentence makes them anomalous for conventional syntactic accounts, their behaviour having to be explained by otherwise unmotivated mechanisms. All that is distinctive about parentheticals from the present perspective is that they constitute separate assertions; as such they have to be of type (t), and I suggest that this simply implies a temporal location. Parentheticals are more flexible in terms of where they can positioned than the other cases considered hitherto. In particular, utterance initial parentheticals are problematic. I outline possible solutions none of which are satisfactory, and this leads me to reconsider how <LINK> operates. I separate it

into the two constituent parts of launching a new tree and identifying a shared node. I consider the possibilities for deriving the most general characterisation of the <LINK> rule. The final formulation not only covers the parenthetical cases, it also allows a more flexible account of the way that structure is derived and interpretation assigned overall.

By extending the theoretical remit of the <LINK> operation I uncover a more detailed account of its properties. My concern is to provide the most generalised characterisation of <LINK> as a process, with other properties stemming from the inherent properties of the elements involved.

In chapter seven I review the results of seeking a unified operation to derive interdependencies, and I draw out the implications. Specifically I consider the way that structure is built according to the revised form of the <LINK> operation that I have developed. I then consider how this might fit into a wider model of utterance interpretation.

Chapter Two

Labelled Deductive Systems for Natural Language: the Formal Machinery

2.0. Introduction

In this chapter I introduce the formal machinery used to build semantic trees in Labelled Deductive Systems for Natural Language, based on the system described in Kempson, Meyer-Viol & Gabbay (in prep.) and developed in other work by those authors (Kempson & Gabbay (1997), Kempson & Gabbay (fthcmng), Kempson & Meyer-Viol (1998)). I start with the tree node predicate, used to specify locations of nodes within the tree. This allows the development of a concept of underspecification for location within the tree, which is used to model phenomena handled elsewhere as movement. I then discuss in greater detail the mechanics of how the structure is built up, specifically the rules which allow development of the tree structure, transitions between states and the annotation of nodes. The lexicon is characterised in terms of which transitions are licensed by particular lexical items and how they create annotations on a node. Having described all the mechanisms required for basic sentence structures, I go through a sample derivation step by step. I then describe the link operation which is used to connect propositions. This is illustrated in the context of relative clauses, for which it was developed. I conclude with an overview of the issues this raises.

2.1. Semantic Trees

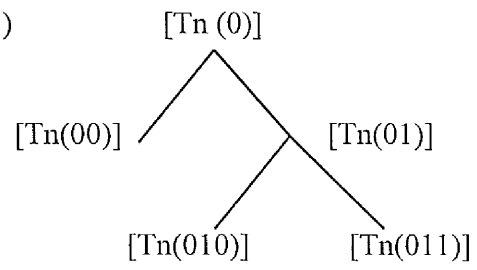
The system models the building of linguistic structure by the development of binary branching trees which indicate the mode of semantic combination: this allows a detailed representation of the stages which constitute the *process* by which such structure is built. Each tree node represents the location of a

Declarative Unit, a premise in the overall derivation, and as such it may be labelled with type and formula information. The actual location of the tree node may be specified by writing in that location, or by graphic representation. At each location the state of the parse is specified in terms of task states: what has already been done by this point and what must still be done according to requirements already established or input from lexical items (see section 2.2.1.).

2.1.1. Tree Node Location

The tree node location predicate, (Tn), varies over values of 0 and 1. These indicate where in the tree a node is located in relation to the root node. (1) below illustrates the specification of tree node locations: Tn(0) is the root node; thereafter (0) indicates a left child, (1) a right child.

(1)



In most of the representations that follow I omit the tree node identifiers for the sake of clarity of presentation.

2.1.2. Tree Node Operators

The tree node logic adopted in this system is the Logic of Finite Trees, LOFT, presented in Blackburn and Meyer-Viol (1994) and further developed in Kempson, Meyer-Viol and Gabbay (in prep.). The two basic modal operators of this system are given in (2)

(2) a. <d> down
b. <u> up

This allows the description of a property as holding at the node either above or below the current location, as shown in (3).

(3) a. <d> P
'some property P holds at the node below the current node'
b. <u> P
'some property P holds at the node above the current node'

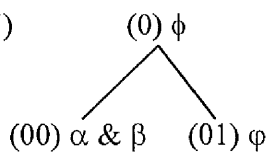
This can be further restricted by specifying the direction, as left (0) or right (1).

(4) $<d>_1$ P
'some property P holds at the node below right the current node'

Note that this is a binary branching system, each split generally representing a step of function application; therefore, while a node may have two children, no node may have more than one parent.

This system allows the description of nodes from the perspective of other nodes. (5) gives the way in which the tree shown can be described from the perspective of each of its nodes using the tree description language.

(5)



At Tn(0) $\{\phi\}$ & $<d>_0$ $\{\alpha \& \beta\}$ & $<d>_1$ $\{\varphi\}$
At Tn(00) $\{\alpha \& \beta\}$ & $<u>_1\{\phi\}$ & $<u>_1<d>_1$ $\{\varphi\}$
At Tn(01) $\{\varphi\}$ & $<u>_0$ $\{\phi\}$ & $<u>_0<d>_0$ $\{\alpha \& \beta\}$

At any location in the tree it is possible to identify any other location. The usefulness of this system is in terms of the mismatch between surface structure,[1] ie the order in which lexical items are introduced to the operation,[2] and the location such an item might have in the final tree according to semantic requirements. This tree description language can express directly both specific,

[1] There is no theoretical significance to this term and none should be assumed. I use it to refer exclusively to the order in which lexical premises enter the system.
[2] By operation I mean the structure building operation, and the processes involved in this. I avoid the term 'parser' for two reasons: i) the system outlined here represents but does not resolve underspecification, and therefore is not a parser in the strict sense; ii) 'parser' implies a distinction between that and the grammar which on current assumptions is at least blurred.

fixed tree locations and comparatively weak descriptions, for example that a node is such that it has some property holding above it.

2.1.3. Underspecification of Tree Node Location

The precise location of a tree node may initially remain unspecified. That is to say, at the point at which the lexical item is introduced to the operation, it itself carries no information as to where in the tree it should be, nor does it match immediately with any existing requirement specified as holding at that point. This is achieved by use of the Kleene star operator, giving reflexive transitive closure. Thus, a specification may be described as holding either at the current location or at a location in some relation to the current node, or at a location in some relation to that recursively, according to the definition in (6), where # is any of the modal operators.

(6) $\langle\#\rangle^* \phi \leftrightarrow (\phi \vee \langle\#\rangle\langle\#\rangle^* \phi)$.

For example, some property may hold either at the current node or at somewhere below the current node.

(7) $\langle d\rangle^* P \leftrightarrow (P \vee \langle d\rangle\langle d\rangle^* P)$
'some property P holds either at the current node or the node below that, or the node below that, etc.'

More generally such a characterisation of an unfixed node will be represented as (8):

(8) $[_* P]$

This operator is local, in the sense that it is confined to the current tree.

When the location of a node is underspecified, the node may be said to be *underlocated*.

The operators given in (9) indicate 'down' and 'up' respectively but these may range over a family of linked trees.

(9) a. <D>

b. <U>

Linked trees are used to set up individuated tasks, which share some node in common, such as relative clauses. The field of application of linked trees and the details of their construction are of central concern to this thesis. They are introduced below in 2.6.

The significance of modelling an underspecified tree location is the way in which this provides an account of the dynamics of those phenomena involving what have been called 'fronted' elements in movement-based syntactic theories. This is dealt with in some detail in Kempson, Meyer-Viol & Gabbay (in prep.). The claim is that utterance initial lexical items may be systematically underspecified for tree node location, and that this is utilised as a strategy for natural language structures.[3] The location of the node is resolved according to subsequent requirements in the development of the tree. I illustrate this with the wh-question in (10).

(10) What does John eat?

Initially, the wh- word remains underlocated; that is to say, at the point at which it is introduced the parsing system can only assign to it a tree node description such that it is located somewhere within the subsequent development of the tree. This is given in (11), which states that at the root node (0) it is the case that the description provided by 'what' is located somewhere below the present node (ie along a chain of child nodes).

(11) $[_0$ ...$[_*$ Fo(WH),Ty(e)]...]

[3] Note, however, that there is no claim of a strict form/function correlation cross-linguistically.

The underspecified node is carried down the tree until such time as the appropriate requirement is there so it can be integrated, and thus the underspecification resolved. I return to the details of this in 2.4.2..[4]

I now turn to the dynamics of how information is built up in the course of tree development.

2.2 The Process of Constructing a Representation

There are two aspects to the building of a representation: building nodes in the first place, and annotating them with the appropriate information. This is effected by the interaction of information supplied by the lexical input with general principles of the parsing schema. The former are discussed in section 2.3; here I introduce the basic structure building operations that the system licenses.

2.2.1. Goals and Task States

The overall goal of the system is to establish a proposition of type (t) with some form of temporal label. As discussed in chapter one, this overall goal is broken down into a number of sub-tasks. These constitute localised tasks, which may be specific to a particular location within the tree. These may be referred to according to the goal at that particular location, hence: e-task, t-task, e→t-task etc. At each node there is explicit indication of the state of the parse at the point reached in the process: ie, what has been done already at that location, what remains to do at that location, what actions, if any, hold still as general requirements on tree development.

This information is represented using the task state indicator. The task state at each node indicates what remains as a requirement TODO (presented to the right of •); and what is a description, which has already been DONE (presented to the left of •). This shows whether or not the local requirements have been

[4] This basic operation of underspecification has been extended to give a unified approach to clitic left dislocation and focus fronting constructions (Kempson & Meyer-Viol (1998)), and to account for variations in crossover effects (Kempson & Edwards (1998)).

fulfilled. (12) and (13) give examples of the use of TODO and DONE respectively:

(12) [Tn(n) • Ty(e)]
'there is a requirement that some formula of type (e) must be instantiated at this location (n)'

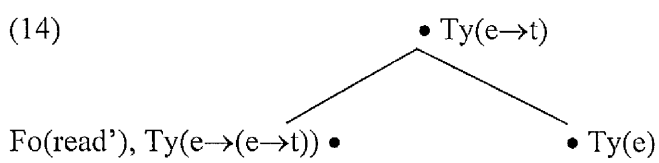
(13) [Tn(n), Fo(billy'),Ty(e)•]
'the formula " billy' " of type "e" holds at this location (n)'
or alternatively
'this location (n) is annotated with the description: formula " billy' " which is of type "e" '

Note that a *location*, (alternatively *node*), may be said to be *annotated* with either a *requirement* or a *description*.[5]

If a requirement or description carries both formula and type information it is said to be a *Declarative Unit*, (DU).

(14) illustrates requirements and descriptions as can be derived in the case of a transitive verb. The description associated with the verb itself is DONE; the subcategorisation requirements of the verb result in the requirement of type (e) TODO.

(14)



Note also in (14) that the top node has the requirement TODO type (e→t). This will duly be fulfilled as and when an object is supplied to fulfil the requirement of type (e) which can combine with the verb to give a description of the requisite type (e→t).

[5] This differs from the terminology presented in Kempson, Meyer-Viol & Gabbay (in prep.).

When all requirements have been fulfilled at a specific location, so there is only description, the bullet may be dropped. Thus, (15) and (16) are equivalent.

(15) $[_n$ Fo(terry'), Ty(e) •]
(16) $[_n$ Fo(terry'), Ty(e)]

These examples also illustrate the general way in which nodes are conventionally written, where square brackets separate off individual nodes. The value of the tree node predicate is written in subscript; so (16) is the same as (17).

(17) [Tn(n), Fo(terry'), Ty(e)]

2.2.2. Transition Rules

The process by which the individual nodes and node annotations are constructed is broken up into a series of steps licensed by rules. These may apply at any appropriate point in the course of the operation and are invoked by lexical specifications. Here I give the general definitions of these rules as stated in Kempson, Meyer-Viol & Gabbay (in prep.), and then a more specific instantiation by way of illustration.

2.2.2.1. Introduction and Prediction

These are concerned with the development of the tree and allow for the introduction of new material.

Introduction (18) breaks down a task into more specific subparts.

(18) Introduction

$$\frac{[_m \cdots \ [_n\ Y \bullet \chi, X] \ldots]}{[_m \cdots [_n \cdots Y \bullet (Y), \langle d\rangle(\psi), \langle d\rangle(\phi)], \ldots]}$$

At a given node (n) specified with the requirement TODO type (Y), it is possible to divide this into two constituent subtasks, specified as restrictions on

the child nodes. This rule allows the introduction of the requirement for pairs of nodes of arbitrary type. (18) gives the rule at its most general.[6]

The most common instantiation of this is from the starting point of the operation, as exemplified in (19). The overall goal TODO Ty(t) is given by axiom; this can then be broken down into requirements of type (e) and of type (e→t) (ie, the goal of deriving a proposition can be broken down into the requirements for a subject and a predicate).

(19) $$\frac{[_n \quad \bullet Ty(t)]}{[_n \quad \bullet \ [<d>Ty(e)], [<d> Ty(e \rightarrow t)]]}$$

Once the child requirements have been established, the rule of Prediction, given in (20), then allows the creation of these new nodes.

(20) Prediction

$$\frac{[_m \ldots [_n \ Y \bullet <j>\phi, X] \ldots]}{[_m \ldots [_n \ Y \bullet <j>\phi, X, [_j \bullet \phi]] \ldots]}$$

where j = {0, 1, *, L}[7]

This advances the operation by building new structure, and the pointer correspondingly moves to the new node.[8]

(21) gives a specific example of how this rule can apply. If a requirement holds at node (n) such that at a child there be a node of type (e→t), then this child node can be created, and annotated with the requirement of that type.

---

[6] This can be restricted to cases of type deduction by the following formulation. In earlier work, this was restricted to type combination operations as given below .

(i) $$\frac{[_n \ldots \bullet Ty(Y)]}{[_n \ldots \bullet Ty(Y), <d>Ty(X), <d>Ty(X \rightarrow Y)]}$$

The more general form in the text is motivated by the need to cover operations other than type combination. These, however, lie beyond the present discussion.

[7] 'L' is the <LINK> relation connecting trees; cf 2.6.

[8] See 2.2.5.

(21)

$$\frac{[_n \bullet \langle d\rangle(Ty(e\rightarrow t))]}{[_{\langle d\rangle} \bullet(Ty(e\rightarrow t))]}$$

The way in which the incremental process of tree growth can be effected by these rules is illustrated in (22).

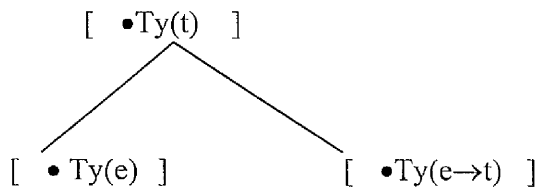
(22) a. Starting point of the parse, requirement TODO Ty(t), ie to derive a proposition

[• Ty(t)]

b. Specify this in terms of sub-tasks, licensed by Introduction.

[• [<d> Ty(e)], [<d> Ty(e→t)]]

c. Build the nodes marked with these requirements, licensed by Prediction.[9]



2.2.2.2. Elimination and Completion

Elimination is the obverse of Introduction, and Completion that of Prediction. These rules concern the compilation of semantic information in the tree, and are basic inference rules which do not increase the overall information state.

Completion allows that an annotation on the child node can be annotated on the parent, moving the pointer back up to that parent.[10] (23) gives the general form.

[9] This shows Prediction building both nodes simultaneously. In the actual derivation of structure from lexical input, this happens step by step. This is shown in the relevant derivations below.
[10] See 2.2.5.

(23) Completion

$$\frac{[_m \cdots [_n Y\ [_j\ \phi] \cdots]}{[_m \cdots [_n \cdots Y, \langle j \rangle\ \phi, [_j \phi]] \cdots]}$$

where $j = \{u, d, L, *\}$

(24) gives a specific instantiation.

(24)

$$\frac{[_n\ (Fo(\alpha), Ty(e \rightarrow t))\ \bullet]}{[_{\langle u \rangle}\ [\ \langle d \rangle (Fo(\alpha), Ty(e \rightarrow t)))\bullet]\ ]}$$

This states that at a node (n) with a completed description of formula (α) and type ($e \rightarrow t$), it is possible to move up to the parent of node (n) and annotate the parent node with the description that at a child node, formula (α) and type ($e \rightarrow t$) holds. This is shown graphically in (25). (The rest of the tree is not shown).

(25) a. The node is marked with the description shown.

$[\ \ Fo(\alpha).Ty(e \rightarrow t)\ \bullet\ ]$

b. This information can be annotated on the node above.

$[_{\langle d \rangle}\ Fo(\alpha).Ty(e \rightarrow t)\ \bullet\ \ ]$

$[\ \ Ty(e \rightarrow t)\ \bullet\ ]$

The general rule for Elimination is given in (26). This applies when the tree description is complete,[11] and allows for the combination of premises.

[11] The rule specifically for cases of type deduction is given below.

(26) Elimination

$$\frac{[_m \cdots [_n \langle d\rangle\psi, \langle d\rangle\phi, Y \bullet X], \ldots]}{[_m \cdots [_n \chi, \langle d\rangle\psi, \langle d\rangle\phi, Y \bullet X], \ldots]}$$

(27) gives the specific example of subject-predicate combination. Once it is established at a node that the child nodes are annotated with the requisite types for type combination, the rule of elimination allows this to proceed. This is equivalent to β-reduction in the lambda calculus.

(27)

$$\frac{[_n \langle d\rangle_0 \, Ty(e), \langle d\rangle_1 \, Ty(e \rightarrow t)]}{[_n \; Ty(t)]}$$

This gives the process of semantic combination when the parsing operation has been completed and all necessary requirements fulfilled. Once parsing of the lexical input in (28) is complete,[12] the rule in (29) can apply.

(28) John runs.

(29) $$\frac{[_n [_{\langle d\rangle} \, Fo(John').Ty(e)], [_{\langle d\rangle} \, Fo(run').Ty(e \rightarrow t)] \,]}{[_n \, Fo((run')(John')).Ty(t)]}$$

Note that nothing in the above rules restricts the occurrence of nodes as either the left or right child.

2.2.3. Thinning

Thinning is concerned with updating the task state according to what has taken place. As discussed above, there may be a requirement TODO something at a

---

(i) $$\frac{[_n \langle d\rangle Ty(X), \langle d\rangle Ty(X \rightarrow Y) \bullet \ldots]}{[_n Ty(Y), \langle d\rangle Ty(X), \langle d\rangle Ty(X \rightarrow Y) \bullet \ldots]}$$

[12] I give a complete derivation in 2.4.

particular node (ie, to the right of the bullet); whenever an action has taken place,[13] this is marked as DONE and the corresponding requirement TODO can be removed. So in (30), the node annotation is updated from being a requirement of type (t) to being a description of type (t). This is effected by the rule in (31).

(30) $\dfrac{[\ \bullet Ty(t)\ ]}{[\ Ty(t)\bullet\ ]}$

(31) Thinning

$$\frac{\{[\ldots[n\ldots\phi\ldots\bullet\ldots\ \phi\ldots],\ n\}}{\{[\ldots[n\ldots\ \phi\ldots\bullet]],\ n\}}$$

2.2.4. Scanning

Scanning is a general term to cover the processes whereby information from the lexicon is incorporated into the structure building operation, effecting some update in the tree description. Information in the lexicon is represented in the form of input/output rules; that is to say, words induce update functions. In the absence of the correct triggering environment, certain default strategies may be employed, the details of which will be addressed as they arise.

The role of templates in the operation is the subject of ongoing research; these may be adopted to capture generalisations about the instantiation of specific constructions[14] and would consist of procedural instructions to undertake tree update operations. Templates determining order of application of update operations might also be used to capture various language specific properties; however, this is a matter for future work.

[13] This may take place according to various processing actions in the system either by free application of the rules or as directed by lexical input.

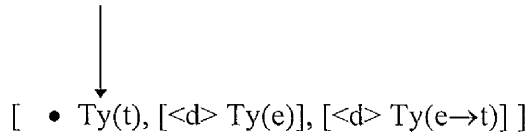
[14] Similar to the approaches adopted by among others Fillmore (1998), Goldberg (1995), Jackendoff (1997), but implemented in a very different way, and with a broader construal of the term 'construction'

2.2.5. The Pointer

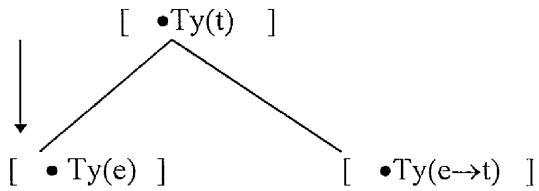
The pointer refers to where the structure building operation has got to in the tree at any particular stage in a derivation, according to the operation of the rules and the instructions from lexical input. Operations are local to the location at which the process is; but that location can be altered by moving the pointer. Movement of the pointer is licensed in two ways: either by steps of Prediction and Completion or by instructions in lexical entries.

As noted above, Prediction allows the building of new nodes; when this occurs the pointer is moved to the new node. (32) illustrates how this works. The arrow marks the location of the pointer; it starts at the root node (a). Application of Prediction builds two child nodes in this case; the pointer may move to either of the two child nodes, as shown in (b) and (c).[15]

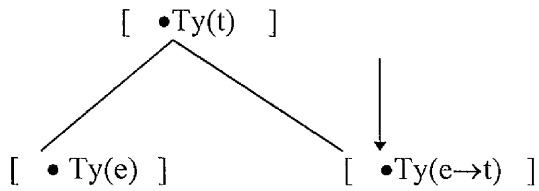
(32) a.



b.



c.



When Completion occurs, the opposite case of pointer movement is taking place. In that case the pointer moves up from the node where the description holds to the parent node, annotating the requisite information on that node. This is shown in (33).

[15] I discuss the linguistic implications of this in 3.6.

(33) a. [•Ty(t)]

[Ty(e→t) •]

b. [[$_{<d>}$ Ty(e→t)] • Ty(t)]

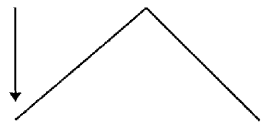
[Ty(e→t) •]

In general, once a node has been annotated with a description, the pointer can return to the parent node. This is further shown in the derivation in 2.4.

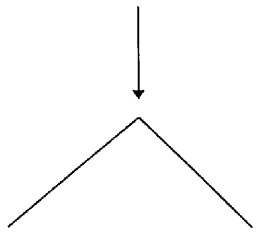
Instructions on pointer movement can be directly encoded in the lexical entry of a particular item. Pointer movement can be written into the lexical entry using the operator π in combination with a certain modality. This is shown in example (34). What this indicates is that from the present location, move up one node, and then move to the right child. This description effects movement from a node to its sibling as shown in (35).

(34) π<u>,<d>$_1$

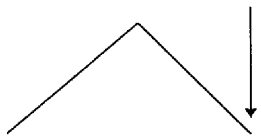
(35) a. Initial Location



b. Intermediate Location, pointer moved up one node



c. Final Location, pointer moved down to the right



Further examples of how this works are contained in the lexical entries below. Lexical information effectively instantiates specific applications of Prediction and Completion. Note then that pointer movement upwards implies the operation of Completion.

2.3 Lexical Specifications

2.3.1. The Form of the Lexicon

Following Kaye (1993) and Jensen (1996) it is assumed that the phonological information associated with a word provides address information, and there is no representation of phonological structure in the lexical entry as such.

The information a word projects can be conceived of as a concept (α) in the form Fo(α) and a specification of its combinatorial properties, its type specification, in the form Ty(x). It is type deduction (generally corresponding to function application over the formula) which builds up conceptual content, where this may also be represented in the form Fo(β).Ty(y). However, this is more correctly understood as a way of describing nodes after the operations of the structure building process. The way in which lexical information is characterised is essentially procedural: that is to say, a word will instantiate a set of procedures concerned with modifying the structure, ie, an update function on a partial tree description.

The primary actions instantiated by lexical entries are i) to build nodes and ii) to annotate these nodes, and functions are defined accordingly. (36) gives the node building function.

(36) Function to build nodes
i
with modality (<u>, <d>, <u>*, <d>*)

Building a node has the effect of moving the pointer to that node. This function effectively enacts lexically specified application of Introduction. Thus the actual process of node building is triggered by the lexical input. The location of the node constructed can, at least in theory, be in any relation to the point at which the input is parsed. A node may be built at any arbitrary distance from the current point.

A node having been built, it can be annotated with information using the functions in (37).

(37) The functions to annotate nodes
  a. f_{DONE} 'annotate with a description'
  b. f_{TODO} 'annotate with a requirement'

These dictate what is to be located there. f_{DONE} will annotate a node with some Declarative Unit information, for example, the type specification and some formula. f_{TODO} will require that some task must be completed at the node. It may also be the case that a lexical item imposes a requirement on some other location within the tree.

Within the process of tree update induced by a lexical entry, the pointer may be moved according to the permutation function π. This has the effect of moving the pointer to the position specified. This navigates the pointer through the structure but does not itself cause structure to be built. Note that when annotations have been made the pointer can return to its initial position according to the rule of Completion.

There is a further sense in which the lexical information is procedural. The way that conceptual information associated with a lexical entry is represented at a node is by annotating the formula description. But what does this formula mean? In terms of the underpinnings of the theory, it is important to bear in mind that this does not represent any external object, as in realist theories, given that the present approach is resolutely representationalist. According to this system, there is no straightforward notion of denotational content for a linguistic expression, nor mapping from a sentence onto its meaning. The entire phenomenon is context bound; and the 'conceptual' part of lexical meaning is

also procedural. It is not that a word maps onto a concept, rather that it gives access to the means of deriving some concept. The content described by some formula Fo(α) represents the concept formation procedure evoked by a lexical item. The details of this are not discussed in the present work.[16]

2.3.2. Lexical Entries for English

In the following sections I give an outline of the form of lexical entries. I should emphasise that at this stage these are provisional in nature, and will be subject to development and modification in the following chapters, particularly in light of the type (p) introduced in chapter three. The lexicon is where a considerable amount of what constitutes the idiosyncratic phenomena of a particular language is located; no implications for languages other than English should be drawn. The precise way lexical entries interact with the overall process of structure building is demonstrated in the derivations in 2.4.

Lexical statements are characterised in terms of inference rules, specifying what actions are to be performed given a certain context. The input state is called the Trigger (Tr), the output is called the Action (Ac). The general schema for presenting lexical entries is shown in (38).

(38) *phonological representation*
'semantic representation'
Tr input state which triggers application of the rule
Ac update actions performed on the tree, which may involve
node building,
node annotation,
pointer movement

2.3.2.1. Transitive Verb

(39) gives the form of a lexical entry for a transitive verb, 'love'.

[16] Such an approach to conceptual meaning is discussed in Carston (1998). Marten (in prep.) analyses the impact of this way of characterising meaning in Labelled Deductive Systems for Natural Language.

(39) *love*
'love'
Tr [•Ty(e→t)]
Ac $i<d>_0$
f_{TODO} (Ty(e→ (e→t)),
f_{DONE} (Ty(e→ (e→t)), Fo(love'))
π<u>
$i<d>_1$,
f_{TODO} (Ty(e))

What this says is that if the current location is a node specified with the requirement TODO Ty(e→t), ie if there is a requirement for a verb phrase: from that node build at the left child a node with the requirement of type(e→(e→t)), then annotate this with the description type(e→(e→t)), formula (love'); move the pointer back up to the starting point, then build a node as right child annotated with the requirement of type (e). Note that the pointer moves to the new node that is built; I do not explicitly write this movement into the rule. I have included the pointer movement from the node of Type (e→(e→t) to the initial node (e→t). This is derivable from Completion, but it is lexically induced. It is the verb which is driving the node building operation, and it annotates the e-node with a requirement which necessitates further lexical input.

According to the specification in (39), nothing should intervene between the verb and the object. The alternative way of describing a transitive verb would be to have a requirement for a child node of type (e). This is the characterisation given in (40). How would these differ in terms of empirical coverage and the strictness of adjacency requirements?

(40) Ac $i<d>_0$
f_{TODO} (Ty(e→(e→t)),
f_{DONE} (Ty(e→(e→t)), Fo(love'))
π<u>
f_{TODO} <d>(Ty(e))

This states that after building and annotating the node of type (e→(e→t), the pointer returns to the point; and that this initial location is annotated with the requirement that it has a child of type (e). While nothing can intervene in terms

of the function application structure, other operations would be permissible in terms of the linear input.

Note the differences between the two characterisations though. The first is much less amenable to flexibility in terms of type mismatching. In this case the node has already been built, and so requires immediate satisfaction of its requirement. Any other lexical material which serves as input to the operation prior to a noun phrase will interfere with this. It is not, however, that further lexical material will cause the derivation to crash. Nor would there be any requirement for reanalysis of already existing structure; the problem is one of the ease with which structure can be built. Consider what happens when an adverb intervenes between the verb and the object, as in (41). Given the structure built by (39), the pointer could move back, but this would be an extremely marked operation, as is indeed the case for English, where (42) is the strongly preferred form.

(41) John smote mightily him.
(42) John smote him mightily.

The second characterisation, (40), does not move the operation forward. Thus, while there is still the requirement for a noun phrase which will be the direct object, if there is a verb phrase adverbial which has to modify the (e→t) node, the operation which the adverbial induces can proceed. The requirement for the node of type (e) would be held in abeyance. This predicts a much greater leeway for insertion of lexical material between the verb and its object. Moreover, this would require a separate application of Prediction, with no apparent catalyst, as noun phrases on their own do not induce nodes. I therefore keep the lexical entry given in (39). Phrase structure as defined over a string becomes a matter of processing in this system, affecting the way that structure is built up incrementally.[17]

2.3.2.2. Entities

[17] What ends up in the final representation is only the outline of semantic relations

Structure building is initially goal driven; further requirements may be specified according to the subcategorisation requirements of the verb. Noun phrases do not themselves induce nodes.[18] They can only create annotations on a node that is already labelled as requiring some entity. The term 'entity' properly is a description of a property which can hold at a node (a term in the DU language). What I am concerned with here are those linguistic items which can fulfil the requirement of an e-task.[19]

Lexical NP

The lexical entry associated with the proper name 'John' is given in (43).

(43) *john*
'John'
Tr [•Ty(e)]
Ac f_{DONE} (Ty(e), Fo(John'))

This states that where there is a requirement for something of type 'e' the following operation will be performed: at that location the formula information (John') will be annotated as a description.

Common Noun

To preserve the requisite semantic formation, common nouns are broken down into a variable and the term predicated of that variable. Crucially, this variable is then available to take part in other linguistic operations such as relative clause construction, as will be seen below. Moreover, the variable itself can be bound by determiners and quantifiers. The form of lexical entry is given in (44).

(44) *man*
'man'
Tr [•Ty(cn)]

[18] The way in which structural operations are induced is, of course, subject to cross-linguistic variation (indeed it forms the locus of explanation for such variation). Unless otherwise stated, my comments apply to English.
[19] The debate over NP versus DP has no content in the present framework given the absence of functional projections.

Ac $i\langle d\rangle_0$
f_{TODO} (Ty(e))
f_{DONE} (Ty(e). Fo(U))
$\pi\langle u\rangle$
$i\langle d\rangle_1$
f_{TODO} (Ty(e→cn))
f_{DONE} (Ty(e→cn). Fo(man'))
$\pi\langle u\rangle$

Determiner

The internal structure of the determiner phrase, as in other theories, is controversial. It is a matter of ongoing research within the framework as to whether or not the interaction between determiner and common noun should be modelled in terms of type deduction. To do so would preserve the basic processes of structure building. However, all other cases of type deduction translate to function application. It is not clear that there is any motivation for term operator binding reflected in this sort of type operation. This is part of the wider debate concerning what semantic operations are required, and how these are manifested lexically and syntactically. Maintaining the standard type-deduction approach gives the sort of lexical entry shown in (45).

(45) *a*
'existential quantification'
Tr [•Ty(e)]
Ac $i\langle d\rangle_0$
f_{TODO} (Ty(cn→e))
f_{DONE} (Ty(cn→e). Fo(ε)),
$\pi\langle u\rangle$,
$i\langle d\rangle_1$
f_{TODO} (Ty(cn)

Application of this induces the need for a common noun. The expression (ε) in the formula introduces the epsilon operator which will ultimately bind the variable to be introduced by the common noun.[20]

While (45) maintains consistency in terms of the actual structure building operation, as noted, term operator binding is obviously a very different

[20] See (Kempson, Meyer-Viol & Gabbay (in prep.) chapter 7.

operation from function application. The alternative is an approach which explicitly does not give the article the same status in the representation.[21] The crux of the problem is what the tree should look like when the semantic operations are not steps of modus ponens. Should the branching operation be restricted to this latter operation? What other tree mechanisms are necessary to represent other semantic functions? The same sort of problem is raised by tense operations. These concerns, however, fall outside the remit of the present work.

For present purposes, these considerations have no bearing on the overall argument. All that is important for the present discussion is that within the cn-task there is a metavariable of type (e). I therefore assume (45) in order to preserve a uniform characterisation of node building within the tree, while noting that in certain cases, specific type-deduction operations lead to specific semantic operations.

Pronoun

As is the case with other noun phrases, pronouns cannot themselves induce the building of nodes, and therefore require the trigger of a requirement of type (e). What distinguishes them from other noun phrases in English is that they encode case information. (46) shows the lexical schema.

(46) *they*
'they'
Tr [•Ty(e)]
Ac f_{DONE} (Ty(e). Fo($U_{pro3pl.}$)),
π<u>
f_{TODO} Ty(t)

The above definition states the following: given a requirement at some tree node location for a noun phrase, annotate that as type (e) with a formula description, the content of which is constituted by a metavariable (U). This formula information represents a search procedure, to locate some entity outside the local domain with which this metavariable is to be identified. Partial information is supplied to specify the entity - in the present case that it

[21] Characterising it, for example, as some kind of label.

represents something which is plural.[22] Move back up to the parent node and annotate this with a requirement that it be of type 't'. In the case of a regular derivation this requirement will be there already and no overall information will be added to the system. However, this guarantees that this lexical item can only occur as the left child of a 't' node.[23, 24]

The same strategy is used in the case of an object pronoun, where the requirement will be that it has a parent annotated with the requirement that it be of type (e→t). This sort of information will vary cross-linguistically in terms of how 'powerful' individual lexical items are, by which I mean the extent to which they can actually *cause* structure to be built. So it may be that case can directly encode the tree node location at which a description holds.

2.3.3. Case as Procedural Information

Case provides information on tree node location: where in the tree the information it marks is to be used. That is to say, the update procedure effected by a lexical item can be specified as occurring in a particular location.

In English, where case marking is relatively infrequent, this is correspondingly given a rather weak characterisation, as outlined above, in terms of marking requirements on parent nodes. However, it is also possible to have situations where case itself is sufficient to invoke the building of a node, regardless of any extant requirement for a node of type(e). This means that case provides a direct characterisation of where in the tree some operation takes place. I have argued elsewhere (Swinburne 1998) that this is how case marking operates in Hebrew.

Note that this is again a very "surface" approach to case; it is not invoked as an underlying property, nor characterised as a necessary part of well

[22] There is a difference between the way that gender and number are modelled, and the way that person is. The former two describe actual properties attributed to the things under discussion. Third person maps onto a different type of search procedure which is actually contextually located. This is not the case for first and second person, which involve a more immediate allocation of identity.

[23] Note that overall well-formedness requirements prevent any node from having more than one parent.

[24] An alternative way to characterise this would be as a requirement on the e-node that <u> Ty(t), but this imposes the problem that the pointer would have to be allowed to move even though all requirements are not fulfilled.

formedness. Either it is present as input information, when it provides information on the structure, or it is absent, in which case the outline of semantic relations has to be otherwise derived. It is positioning within the semantic tree which determines interpretation, regardless of how such structure has been derived.

2.4 Sample Derivation

Having introduced the mechanisms whereby structure is built up, I am now in a position to illustrate these by going through a sample derivation. The approach outlined here adopts the general idea within Relevance Theory (Sperber and Wilson 1986/95) that the automatic goal of human cognition when exposed to verbal stimuli is to attempt to derive contextual effects, the initial step of such a process being to establish some proposition. Within the current conception of the linguistic system, this is instantiated by the goal of establishing a proposition with a temporal index; that is, by having a requirement TODO type (t) annotated on the root node of a tree, location (0).

(47) • Ty(t)

This is always the starting point of the operation, and reflects system-internally the way in which the operation is weakly goal directed.

The rule of Introduction allows this to be respecified in terms of the subtasks TODO type (e) and TODO type (e→t) holding as sibling nodes, children of the current node.

(48) • Ty(t), $[_{<d>}$ Ty(e)], $[_{<d>}$ (Ty(e→t)]

Prediction is freely available to drive the construction of nodes. This allows the building of the subject position shown in (49).

(49) $_{(0)}$ • Ty(t), [$_{<d>}$ (Ty(e→t)]

$_{(00)}$ • Ty(e)

Now, given the input string in (50), the assignment of structure can proceed incrementally. 'John' is scanned, and the lexical information is accessed that for any update to occur the pointer must be at a node of requirement type (e). In the absence of any other intonational indicators, the formula (John') of type (e) will duly be inserted at tree node(00), according to the lexical entry shown in (51).

(50) John loves Bill.

(51) *john*
'john'
Tr [•Ty(e)]
Ac f_{DONE} (Ty(e), Fo(John'))

The tree is updated to (52).

(52) $_{(0)}$ • Ty(t), [$_{<d>}$ (Ty(e→t)]

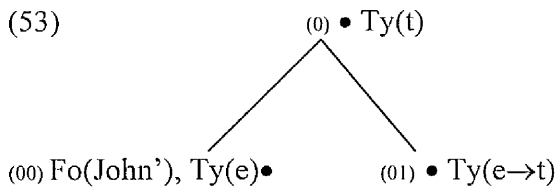
$_{(00)}$ Fo(John'), Ty(e)•

When this subtask has been fulfilled, the pointer returns to tree node (0), according to Completion.[25] Prediction now allows the construction of a new node according to the requirement holding at tree node (0), to give the structure in (53).

[25] Strictly speaking, this updates tree node (0) to (i). I omit the completed description to keep the presentation clear, and as this can be seen from the tree as a whole.

i. [$_0$ [<d> Fo(John).Ty(e)] • Ty(t), [<d> Ty(e→t)]]

(53)

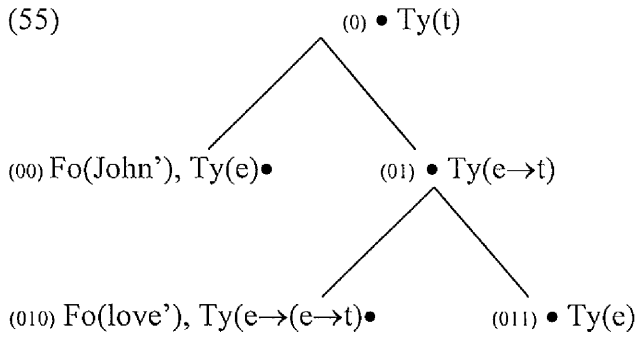


The next lexical item to be scanned is the verb 'love'. The requirement at tree node (01) is for type(e→t). This is the correct trigger for the actions invoked by the verb, according to the lexical entry in (54).

(54) *love*
'love'
Tr [•Ty(e→t)]
Ac $i\langle d\rangle_0$
f_{TODO} (Ty(e→ (e→t)), Fo(love'))
f_{DONE} (Ty(e→ (e→t)), Fo(love'))
$\pi\langle u\rangle$
$i\langle d\rangle_1$, f_{TODO} (Ty(e))

The result of the update achieved by this lexical entry is the structure shown in (55).

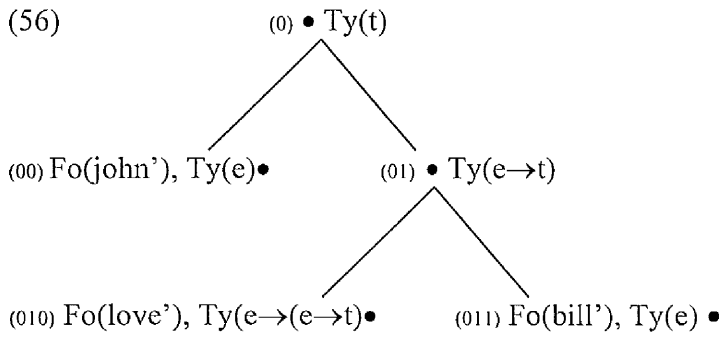
(55)



Now there remains the requirement of type (e) at tree node (011).

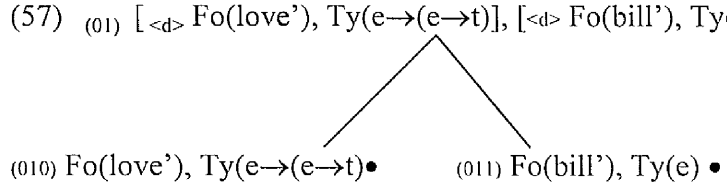
Scanning the next lexical item, 'Bill', duly provides the right input for the operation, and the lexical update effected gives the structure in (56).

(56)



All the subcategorisation requirements have been met, and compilation of the tree can proceed.[26] Completion at tree node (01) gives the representation shown in (57).

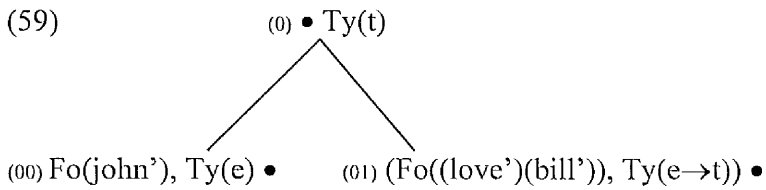
(57) (01) [<d> Fo(love'), Ty(e→(e→t)], [<d> Fo(bill'), Ty(e)] • Ty(e→t)



Application of the rule of Elimination to (57) gives (58), which (59) shows in the wider tree structure.

(58) (01) (Fo((love')(bill')), Ty(e→t))

(59)



Applying Completion to (59) gives (60).

(60) [$_0$ [<d> Fo(john'), Ty(e)], [<d> Fo((love')(bill')), Ty(e→t)] • Ty(t)]

Elimination gives the semantic formula in (61), a proposition of type (t) where all requirements have been met.

[26] The term 'compilation' is used to describe the process whereby steps of Completion and Elimination are undertaken to derive the ultimate meaning.

(61) (0) Fo(((love')(bill'))john), Ty(t)

The above outlines the basic process of constructing an interpretation for subject verb object.

I now return to the issue of nodes which are initially underlocated. I discussed the underlocation of nodes in 2.1.3. I consider the wh- question in (62).

(62) Who does John love?

(63) gives the lexical entry for 'who'.

(63) *who*
'WH human'
Tr [•Ty(t)]
Ac f_{DONE} Cat(Q),
i<*>
f_{TODO} (Ty(e))
f_{DONE} (Ty(e). Fo (WHO))

This states that given a requirement of type(t), the node can be annotated with the category information (Cat) that it is a question (Q).[27] Then an underlocated node is built which is of type (e) and which contains a place-holding metavariable, with the information '+ human'.

The tree update effected by this is shown in (64). (Assuming that no other information has yet been received as input).

(64) •Ty(t); [* Fo(WHO).Ty(e) •]

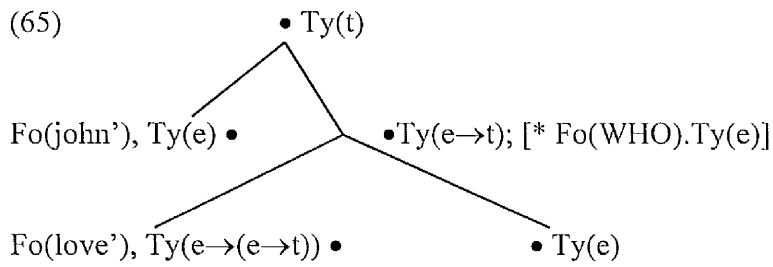
Here I adopt a semicolon after the requirement to separate off the underlocated node from other requirements and descriptions that may be annotated on a node. The underlocated node is itself DONE in terms of its Declarative Unit information, as indicated within the square brackets; it is not a requirement on

[27] See Kempson, Meyer Viol & Gabbay (in prep.) for details of this.

the current location, so it should not be conjoined to the annotated requirement using the comma. However, in terms of the tree development as a whole, it still requires some operation to take place, namely that its underlocation be resolved. This notation is used throughout the rest of the thesis.

What happens next in building structure for (62) is that the process continues until all the lexical input has been used. The structure derived at that point is shown in (65).

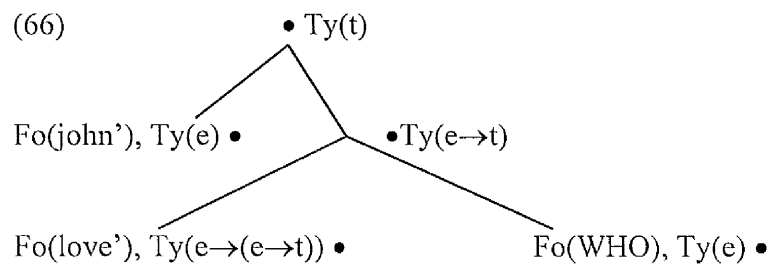
(65)



The underlocated node built by 'who' has been 'passed down'.

The requirement for a node of type (e) (as object) is projected by the verb; without something to fill this node the derivation will crash. The underlocated node is also of type (e), and therefore matches this requirement. The underspecification of its tree node location can duly be resolved here, giving the overall tree structure in (66).

(66)



All the requirements have been fulfilled and the compilation takes place as normal.

I have outlined above how structure is built up incrementally according to instructions encoded in lexical input and the general principles of the system. Underlocation is resolved by matching with requirements holding independently

in the tree. These basic principles will be invoked throughout the rest of the thesis.

2.5 Adjunction

The operations outlined above map the process whereby structure is built up in the case of basic predicate-argument relations, where a verb combines with a subject and (optionally) an object. Following general semantic considerations, this is taken to be the basic unit, a truth-evaluable proposition, consisting of an entity and a predicate.[28] This leaves an array of natural language data which do not obviously fit into this pattern, and the problem of how to deal with these in the current theory. A number of approaches have been proposed to the problems of adjunction phenomena. I should stress, though, that the term 'adjunct' has no theoretical content in the present discussion and is used purely as a term of convenience to describe a range of phenomena which, seemingly, supply non-essential information.

The dividing line between adjuncts and arguments has concerned optionality: whether, for example, a noun phrase is necessary to fulfil the subcategorisation requirements of a verb. Goldberg (1995) argues that this approach to subcategorisation is seriously flawed on the following grounds: it requires vast lexical entries to capture valency alternations, and therefore postulates enormous areas of redundant information, failing to capture the generalisations across these. She argues that such generalisations can be explicated in terms of constructions.

Marten (in prep.) breaks down the argument/adjunct distinction by proposing a system of type underspecification in the verb phrase. Adopting the type specification $(e^* \rightarrow t)$ as the general characterisation of verbs enables the incorporation of additional arguments into the verb phrase. This is defined thus:

(67) $e^* \rightarrow t = (e \rightarrow t) \vee (e^* \rightarrow (e \rightarrow t))$ recursively.

[28] In order to be truth evaluable, a proposition also minimally needs some kind of temporal location. The details of this are not pertinent here.

Marten (op cit). makes a substantive claim about the nature of the syntactic information, namely that it severely underdetermines the output, in that it does not fully define the argument structure. That is to say that the logic gives an underspecified characterisation of the combinatorial possibilities associated with any particular verb; the number of arguments that can be incorporated will depend upon on-line concept construction. This implies that there are not preordained lexical entries with valency alternations; but rather there are dynamic concept construction procedures reflecting the lexicon as an interface between linguistic knowledge and more general knowledge stores, but lacking what might be termed a 'dictionary' of classical definitions.[29] In this respect there are similarities to the approaches outlined by work in Goldberg (1995) and Carston (1998). The incorporation of additional arguments may be licensed by case marking on noun phrases or prepositions in combination with noun phrases, to indicate how complex relations are built up. The details of this approach are given in Marten (in prep.), to which the interested reader is referred.

My concern, however, is the development of adjunct structures which cannot be incorporated into the basic pattern of function-argument operation, ie, adjuncts which do not form a complex unit either with or within the verb phrase. What sort of operations are definable to capture the contribution made by siuch adjuncts, both in procedural terms and in terms of the ultimate denotational content? To what extent is this constrained by lexical specifications as opposed to the general dynamics of the operation? Can the framework provide a coherent account of secondary predication?

These questions form the subject of the rest of the thesis. Specifically, I examine the modification of noun phrases by additional predicates, and how this integrates into a general model of structure building for an utterance. To examine this I need to introduce a further part of the LDSNL approach first

[29] In this cognitive approach, there are no necessary and sufficient conditions to define concepts, nor do words have definitions such that these have to have been learnt in order for a word meaning to have been acquired. Rather there are context dependent conditions on felicity of usage according to the contextual effects the speaker is aiming to achieve.

proposed in Gabbay and Kempson (1992), and developed in Kempson, Meyer-Viol & Gabbay (in prep.), namely the <LINK> operation.

2.6 The <LINK> Operation

2.6.1. Introduction

The <LINK> operation allows for the disruption of the tree building process at a particular node location, and the commencement of a new tree with root node annotated with the requirement of type (t). This, like the other rules, may be optionally applied. By definition, a well-formedness requirement holds on the structural process such that: the description holding at the node from where the link tree is launched has to hold also somewhere within that tree. The ramifications of this will be discussed below. The semantics defined for <LINK> within Kempson, Meyer-Viol & Gabbay (in prep.) are of simple co-ordination. This also will be developed in the rest of the thesis in terms of impact on both the procedural and structural characterisation of <LINK> derivations, and in particular the linkage of non-equivalent structures.

By hypothesis, instantiations of the <LINK> operation will be found universally as a property of natural languages. The extent to which it is freely available as a parsing strategy, constraints on its application, and its association with particular constructions or lexical items across languages all remain open questions.

2.6.2. The <LINK> Rule

(68) gives a generalised form of the <LINK> rule.[30]

(68) <LINK> Rule (General Form)

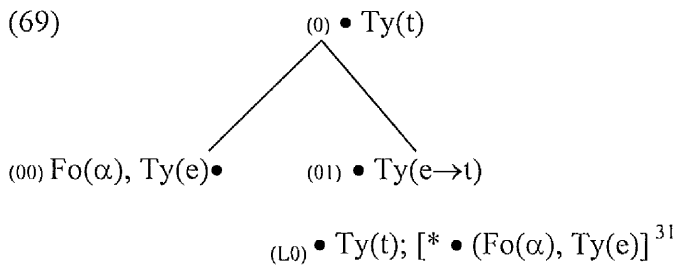
$$\frac{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] .. \bullet \, Ty(t)]}{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] .. \bullet \, Ty(t)], [_{nL} \bullet \, Ty(t), [_* \; \bullet (Fo(\alpha), Ty(e))]]}$$

[30] As throughout this chapter, the definitions for LDSNL have been developed in various works by Kempson, Gabbay and Meyer-Viol. The rules given here are adapted from Kempson, Meyer-Viol & Gabbay (in prep.), which gives a family of <LINK> rules for relative clause phenomena.

Within a tree defined by a requirement of type (t), at a node annotated with a description of type (e), it is possible to start a new tree described as linked to the existing tree, on condition that the description holding at the original node is carried over as a requirement TODO in the new tree. Note that imposing a requirement on a tree necessitates lexical input to the operation to fulfil that requirement.

(69) shows how the <LINK> operation is launched from tree node (00); this results in a new tree of requirement type (t), which also has as a requirement the description from tree node (00).

(69)


[31]

(70), below, gives the rule for elimination of <LINK>. This can apply when all the requirements in both trees have been fulfilled.

(70) Elimination of Link

$$\frac{[_0\ Fo(\alpha), Ty(t)],\ [_{L0}\ Fo(\beta), Ty(t)]}{[Fo(\alpha), Ty(t)\ \&\ Fo(\beta), Ty(t)]}$$

I now turn to the specifics of how <LINK> applies in natural language.

[31] Strictly speaking, the tree node location here should be 00L0, in that the node to which this tree is linked is 00. However, as this is made clear by the shared DU information, I generally omit this for the sake of clarity of presentation. Whether or not this will be a suitable means to distinguish the location from which the link tree is initiated from the semantically linked node (ie that node which is shared between the two trees) in cases where there is not strict adjacency will be discussed below.

2.7. <LINK> and Relative Clauses

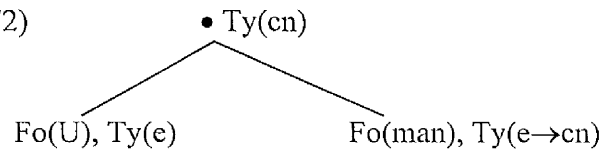
As defined above, the description of the node which is carried over is passed across as an unfixed requirement into the <LINK> tree (henceforward the L-tree). For the emergence of a well-formed structure, there are *two* stages of resolution here which need to be clearly distinguished. Firstly, the location within the tree (of this requirement) is unfixed. Secondly, there is the fact that this is only a requirement and that, ultimately, all requirements must be fulfilled. That is to say, this imposes the need for further material from the lexicon.[32] There is no reason to suppose that languages should be uniform in the steps by which this is achieved, and as discussed in Kempson, Meyer-Viol & Gabbay (in prep.), this provides a major source of typological variation. These two factors provide the basis for the investigation of relative clauses.

2.7.1. The Complementiser as a Relative Pronoun

The first strategy I examine is that displayed in 'full' relative clauses in English, where by 'full' I mean tensed clauses headed by a relative pronoun as in (71).[33] The <LINK> rule is freely available,[34] so the point reached in the building of structure at which the relative clause enters the operation is as shown in (72). This gives the internal structure of the noun phrase (and omits all structure not under immediate consideration).

(71) the man who John loves took a taxi

(72)



The first problem to consider is how the <LINK> rule can apply. The rule states that at a given node in the tree, it is possible to launch a new linked tree. By

[32] Needless to say, this initial characterisation is provisional, and will be modified subsequently.
[33] This is a restrictive relative clause, and so modification takes place within the common noun, in other words, at the equivalent of some N-bar level.
[34] What causes application of the link rule will be discussed presently.

definition the linked tree must somewhere contain the node from which the operation was launched. Moreover that node has to be of type (e). This raises the question of where the pointer would be located, and whether any backtracking operation is involved.

It is not possible to launch a link tree from the node annotated as type (e→cn), nor is it possible to do so from the node with the requirement of type(cn). For the <LINK> rule to apply the pointer has to be at the node annotated Fo(U), Ty(e). However, as it is currently set up, the lexical entry for a common noun will return the pointer to the node annotated Ty(cn). I therefore modify the lexical entry for common nouns to the characterisation given in (73).

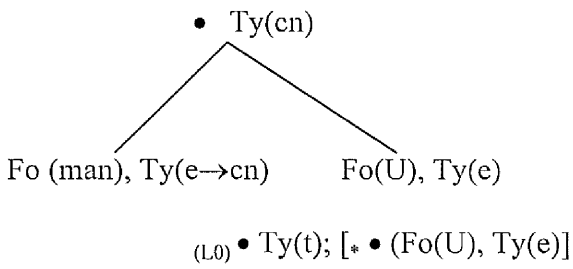
(73) *man*
'man'
Tr [•Ty(cn)]
Ac $i<d>_0$
f_{TODO} (Ty(e→cn))
f_{DONE} (Ty(e→cn). Fo(man'))
$\pi<u>$
$i<d>_1$
f_{TODO} (Ty(e))
f_{DONE} (Ty(e). Fo(U))

This has been modified in two ways. The restriction is built before the metavariable, switching these in terms of left and right, and the final pointer movement has been omitted.

The switch between left and right effected by the above change has no bearing on the interpretation but does allow the operation to proceed straightforwardly. The pointer is not automatically moved back up to the node of type (cn). Completion is left as a free option when the lexical instructions have been completed.

If the common noun is defined thus, the derivation will proceed as shown in (74), which gives the state of the process after the <LINK> rule has applied.

(74) After application of the <LINK> rule



At this stage the L-tree is annotated with an underlocated requirement for the metavariable.

What the relative pronoun does then is to satisfy the requirement in the underlocated node, so that the content of this node becomes a description, rather than a requirement. In this sense, it functions merely as a channel and does not itself contribute any declarative content as such. This is in keeping with the general function of wh- words, which is to supply a metavariable. The wh-word, moreover, does not provide any further information about where in the tree the node is located. (75) gives the general update associated with a wh-relativiser.

(75) Lexical entry for wh- relativiser

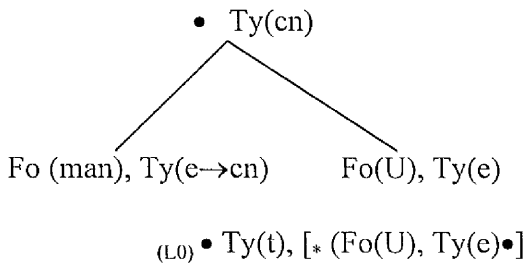
WH Tr $[_{nL}\ [_{*}\ \bullet\ (Fo(\alpha), Ty(\gamma))]$

Ac $f_{DONE}\,(Fo(\alpha), Ty(\gamma))$

In the case of 'who', the type is restricted to type (e) and there is the further stipulation that the formula with which it identifies has to represent something human (or at least animate).

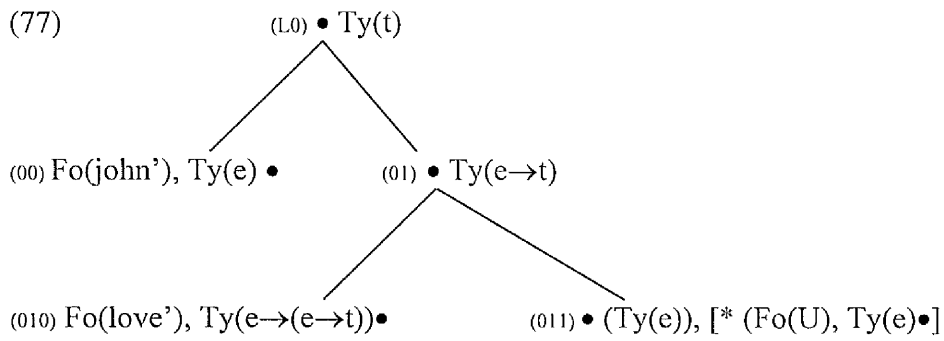
(76) shows how the tree is updated as a result of processing 'who'.

(76) After scanning of 'who'



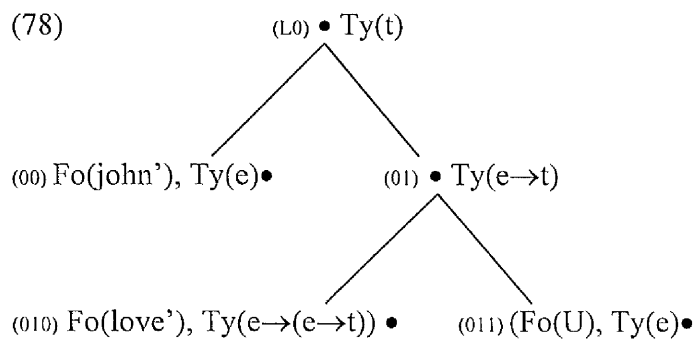
The rest of the derivation proceeds regularly with the requirement that somewhere in the tree the underspecification of the node location has to be resolved. (77) illustrates the state of the operation in the <LINK> tree when all the lexical information has been scanned.

(77)



The underlocated node matches the requirement at (011) and can therefore be instantiated there, to give the completed L-tree in (78).

(78)



Completion and Elimination take place to compile the tree in exactly the same way as happens in an 'ordinary' tree. The pointer can then return to the initial tree, and all tasks within the 'e' node having been completed, the rest of the derivation for that tree will continue.

This strategy first guarantees that the requirement is transferred into a description and then treats it in the same way as any other underlocated expression. Extraction patterns are the same in relative clauses and in wh-questions.

(79) who came?
(80) the man who came

(81) what did you see?
(82) the sheep which you saw
(83) what did you put the pizza in?
(84) the oven which you put the pizza in
(85) where did they hide?
(86) the cave where they hid

As alluded to above, it is not the case that all languages will adopt the same means to deal with the problems of processing <LINK> structures. Kempson, Meyer-Viol & Gabbay (in prep.) claim that in Arabic the relativiser is an expletive element; the requirement carried over remains a requirement, and hence requires further lexical input in the form of a resumptive pronoun which then both fixes the tree node location and makes the requirement a description. Hebrew is claimed to be a mixed strategy language where the Declarative Unit information can be carried over either as a requirement or as a description. For the present, however, I concentrate on the details of English.

2.7.2. The Dynamics of Restrictive versus Non-Restrictive Relatives

The example given above is of a restrictive relative clause. The original starting point for the development of the <LINK> analysis given in Kempson, Meyer Viol & Gabbay (in prep.) was the problems raised by non-restrictive relative clauses. A number of differences between the two types of structure have been noted, primarily related to the degree of syntactic connectedness between the head and the relative (cf Ross 1973, Emonds 1976, 79, Jackendoff 1977, Safir 1986, Fabb 1990). For example, non-restrictives do not display cross-over effects, nor are negative polarity items licensed, as the examples below illustrate.[35]

(87) The man$_i$ who his$_{*i}$ mother hates arrived yesterday,
(88) The man$_i$, who his$_i$ mother hates, arrived yesterday.
(89) No student who had any sense would go there.
(90) *No student, who had any sense, would go there.

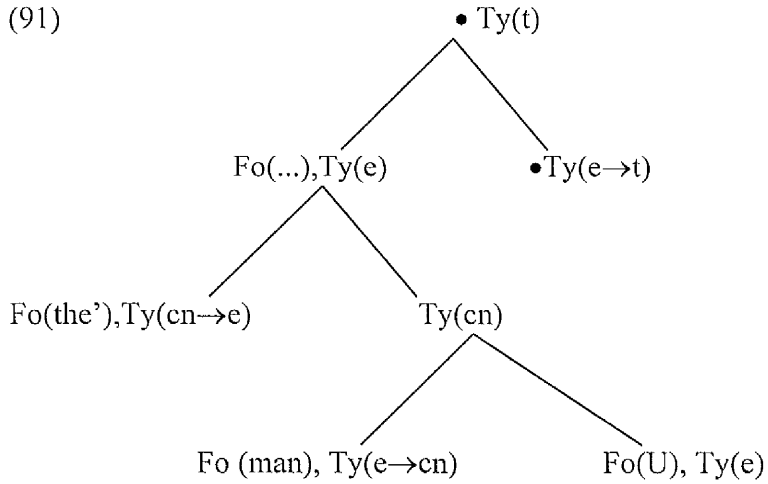
[35] There is not general agreement that restrictive relatives do exhibit crossover; however, non-restrictives never do.

The general approach adopted has been to postulate another level of syntactic representation where the co-indexing operation between a non-restrictive and its head takes place. Safir (1986) has a level of LF'. Fabb (1990) differentiates between the syntactic operation of predication which occurs at LF in the case of restrictive relatives and a looser 'aboutness' relation characterising the relation contracted with the non-restrictive relative at a discourse level called X-structure. In neither case, however, is it apparent what motivates such an additional level more generally within the theory, nor is it clear whether it should be considered a level of syntax proper.

The approach to natural language interpretation adopted here claims that there is a single level of representation, but a dynamic process of deriving that. The differences between restrictive and non-restrictive relatives are captured in the stage at which the <LINK> operation is induced, and therefore what descriptive unit is shared by the main tree and the L-tree.

Recall that in a restrictive relative what is passed over is the variable (U) which is supplied by the common noun predicate. Thus the launching of the L-tree operation occurs *within* the overall e-task. What is derived is a further specification of the properties attributed to this variable. Any quantifier binding operations will occur over both instantiations of the variable. Similarly, licensing operations, such as negative polarity, will have access to *both* predication operations, ie the relative clause and, vacuously, the common noun predicate. The overall task requirement, to construct a Declarative Unit of type 'e', is not fulfilled until the L-tree has been built, all premises having been used, and the process completed at the original node. At this point the entity is established, and so the operation of structure building can proceed for the initial tree, as illustrated in (91).

(91)



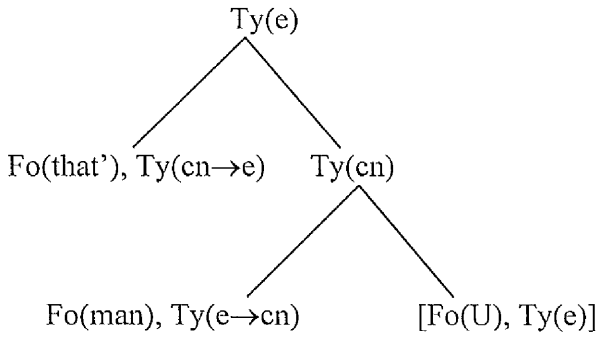
$_{(L0)}$ [(Fo((loves(U))(John)), Ty(t)]

In the case of a non-restrictive relative, the Declarative Unit information that is carried over into the L-tree is that of the e-node overall. That is to say, it is only when that e-task has been completed that the <LINK> operation takes place.

(92) that man, who John loves, is the new anthropology lecturer.

The structure building operation for (92) is guided by the intonation: the pause which invokes the non-restrictive interpretation indicates that the current (e-) task has been completed. Compilation takes place within the node; the pointer returns to the top of this task and it is the whole annotation which is carried over into the L-tree. (93) shows the node at this stage - note that all the requirements have been fulfilled.

(93)



(94) shows the initial structure after the <LINK> operation; the whole description associated with the e-node in the initial tree is carried over as a requirement. The relativiser 'who' turns this into a description, which is still underlocated (95).

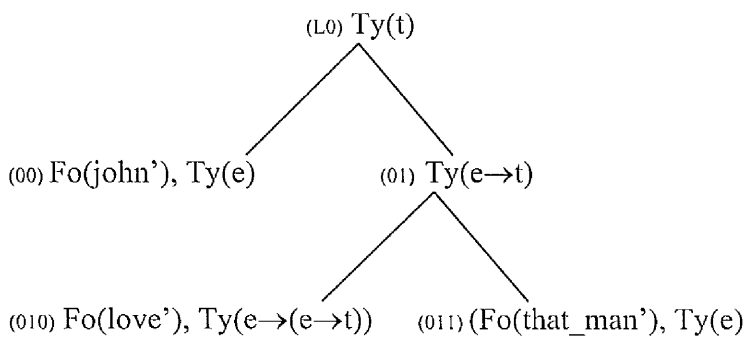
(94) Starting point of L-tree

$$_{(L0)} \bullet \text{Ty(t)}; [_{*} \bullet (\text{Fo(that\_man')}, \text{Ty(e)}]$$

(95) After scanning 'who'

$$_{(L0)} \bullet \text{Ty(t)}; [_{*} (\text{Fo(that\_man')}, \text{Ty(e)}]$$

(96) After processing of all lexical input:



This explains the impossibility of syntactic interaction between the relative clause and the head; the head has become fully contained within the relative clause (notwithstanding its independent existence in the main tree). In terms of the stages in which the structure is built up, the e-task has been completed, and the operations of Completion and Elimination have taken place. It is therefore closed off and the L-tree has no reference to the internal structure of that node.

2.7.3. Relatives without Complementiser

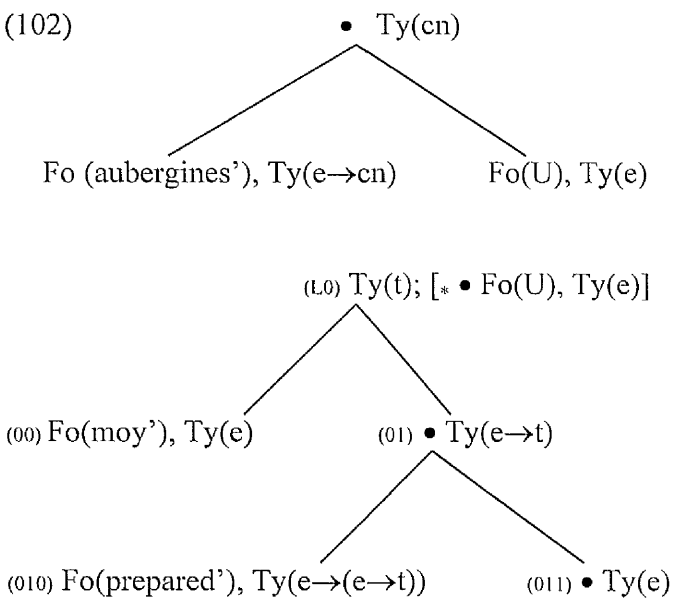
So far I have only examined cases where there is an overt wh- pronoun, and it is this which transfers the requirement into a description. In English, tensed relatives with a null relativiser are acceptable when the shared element is in non-

subject position. Thus, in contrast to (97b), the (b) examples in (98)-(101) are all perfectly grammatical.

(97) a. the man who came in sat down.
b. *the man came in sat down.
(98) a. the woman who I saw yesterday comes from Sheffield.
b. the woman I saw yesterday comes from Sheffield.
(99) a. the aubergines which Moy prepared were delicious.
b. the aubergines Moy prepared were delicious.
(100) a. the table which I put the bag on
b. the table I put the bag on
(101) a. the club which we went to last night
b. the club we went to last night

The present form of the <LINK> rule cannot account for these, as in the absence of the relativiser, there is nothing to fulfil the requirement. That is to say, the variable will be carried across as an underlocated requirement, but there will be no lexical input to make this a description. Consider the structure building process for (99b), as shown in (102), (ignoring the details of the determiner and the rest of the main tree).

(102)



There is the requirement for the formula (U) of type 'e' at some point in the tree. Moreover, at tree node (011) there is a requirement for an entity. Both of these

are *requirements*; and so, in the absence of further lexical input, the derivation would crash.[36]

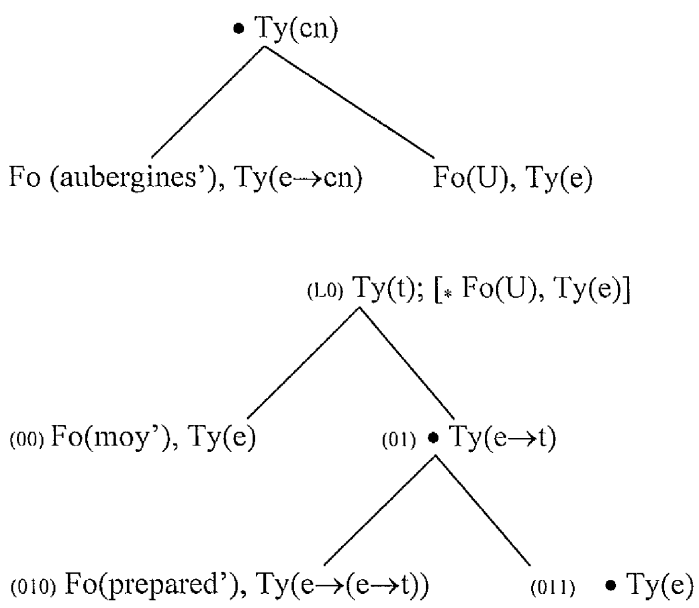
Of course these examples are perfectly acceptable, so Kempson, Meyer-Viol & Gabbay (in prep.) develops a second <LINK> rule to account for these cases.

(103) <LINK> Rule Without Relativiser

$$\frac{[_m \ldots\ldots\ldots [_n Fo(\alpha), Ty(e)] .. \bullet\ Ty(t)]}{[_m \ldots\ldots\ldots [_n Fo(\alpha), Ty(e)] .. \bullet\ Ty(t)], [_L \bullet\ Ty(t);\ [_* Fo(\alpha), Ty(e))]]}$$

This states that when the <LINK> operation is launched, what is carried over can be added to the L-tree as a description, rather than as a requirement. This then removes the need for any further lexical input, as this is a description and not a requirement. The only need now is to resolve the underspecification of the tree node location. (104) shows how the node can now be carried over as a description.

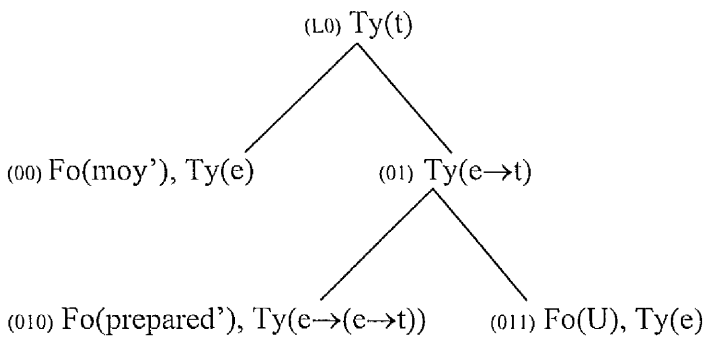
(104) After Applying the <LINK> Rule



[36] This is to adopt familiar terminology to reflect the absoluteness of the logical operation. What this means is that no well-formed structure can be derived.

To resolve the underlocation of a Declarative Unit which is a description, what is required is that there be a *requirement* for something of that type. This is exactly the situation at tree node (011), so this is where the underlocation is resolved.

(105) After resolving underlocation by fulfilling requirement



The general pattern is that a description will be carried down a tree until such time as there is the correct requirement.

This account has major empirical consequences for resumptive pronouns. The prediction is that if a language has the '<LINK> without relativiser' rule, then it will not have compulsory use of resumptive pronouns as a regular part of the grammar. If it does not have this rule, then resumptive pronouns will be necessary.

Recall that there were two problems in deriving a well-formed L-tree. The first was the status of the Declarative Unit passed over, as this has to end up as a description. The second was the location of this Declarative Unit. In Arabic, both questions are solved at once by the use of a resumptive pronoun. In English, a relativiser sorts out the first problem, but the second is resolved by the dynamics of tree building by means of the possibility of incorporating this Declarative Unit into a requirement marked on the tree. In fact, the same strategy is employed when there is no complementiser, and the Declarative Unit is passed over as a description.

The introduction of this rule to allow complementiser-less relativisation raises a number of related questions. What motivates the existence of two rules

in English? In other words, if the Declarative Unit can sometimes be carried over as a description, then why can it not always be carried over as a description? What is the point of having available two strategies? The existence of complementiser-less relatives shows that the <LINK> rule implementing the 'description' strategy is definitely necessary for English. Assuming the minimal hypothesis that this is the only link rule, what problems would arise? The major empirical stumbling block are tensed relative clauses where the shared Declarative Unit is the subject of the relative (106). In this case, the presence of the relativiser is compulsory. Yet nothing should rule out the possibility that the node be located in the L-tree as subject without further lexical input, as is indeed the case for participial relatives (107).

(106) a. The man who conducted the orchestra was Rostropovich.
b. *The man conducted the orchestra was Rostropovich.
(107) The students sitting in the corridor study anthropology.

The obvious explanation to give here is a processing one: that other things being equal, input of a noun phrase plus a tensed verb will be parsed as a main clause, and that reanalysis is not available. The wh- relativiser is therefore necessary to indicate that an L-tree should be constructed. This would make wh- relativisers in English equivalent to expletives,[37] anticipatory elements contributing nothing to the final representation.[38] Though this has some plausibility, the violation here seems more absolute - it is not just that building a <LINK> structure is unmotivated, but that it is impossible for the given input.

Intuitively, it does seem that the WH-relativiser is in some sense necessary to provide a structural position. So, the assumption that there is only one <LINK> rule for English still leaves to be explained the problem that there has to be a relativiser for the subject position in a tensed clause.

This overall picture is further confused by considering non-restrictive relative clauses. Non-restrictive relatives display different behaviour from restrictives in

[37] In the sense of Kempson (1998).
[38] Though a distinction can be drawn between an expletive element with no semantic content whatsoever, and one which does give information such as + human, but where this is given no structural representation.

terms of the distribution of relativisers. In the examples below I use a proper name in order to guarantee the non-restrictive interpretation.

(108) a. Terry, who works in the City, has an office near the Barbican.
b. *Terry, works in the City, has an office near the Barbican.

(108) shows that with regard to the requirement for a relativiser for the subject position, non-restrictives and restrictives pattern identically. However, with regard to object and adjunct positions, non-restrictives always require the presence of a complementiser in tensed clauses, as (109) and (110) show (compare (98)-(101) above).

(109) a. Tony, who I saw yesterday, comes from Sheffield
b. *Tony, I saw yesterday, comes from Sheffield
(110) a. Tony, who Karl visits museums with, comes from Sheffield
b. *Tony, Karl visits museums with, comes from Sheffield

A potential source of explanation would of course be the nature of the declarative unit that is passed over, as this differs in the two cases. Recall that in the case of the restrictive it is a variable, whereas for a non-restrictive it is the result of the completed e-task. However, it is the latter case which is, in current terms, more complete and hence, if at all different, the theory would suggest that *these* should not require the support of a pronoun. Furthermore, it is possible to have non-tensed, non-restrictive relatives without having a relativiser.

(111) Terry, sitting at the edge of the bar, ordered himself another Campari.[39]

This demonstrates that it cannot be the content of the Declarative Unit which is responsible for this difference.

The difference in behaviour between strategies involving transfer of a Declarative Unit as an annotation and transfer as a requirement was originally invoked to explain the cross-linguistic variation in crossover phenomena and the

[39] These are generally termed parentheticals. In the same sense, non-restrictive relatives are parentheticals. Both are derived by the <LINK> operation. These are discussed in chapter six.

distribution of resumptive pronouns. Invoking two link rules for English will capture the facts, but does not explain why the difference in strategies should come about in the way that it does. Adopting this approach makes the optionality of the relativiser for non-subject tensed restrictives a mystery. The widespread availability of the <LINK> strategy in English, which will be detailed below, suggests that this may be a problem specific to that environment.

As defined, the <LINK> operation does not require lexical triggering. The wh- relativiser turns a requirement into a description, but this is not generally necessary. In English at least, nodes can be carried across as (underlocated) descriptions. Effectively, the Declarative Unit is being used 'twice'. This opens up the possibility for extending the <LINK> analysis to other cases which apparently require the instantiation of syntactic elements with no surface form. Whether this is indeed the case, how the process of deriving that derivation proceeds and what effects this has on the interpretation will be examined in the following chapters.

2.8. Conclusion

In this chapter I have been concerned with the details of deriving interpretations using Labelled Deductive Systems for Natural Language, based on the system presented in Kempson, Meyer-Viol & Gabbay (in prep.). This adopts a concept of semantic trees, each node annotated with a Declarative Unit of type and formula information. This can be annotated as TODO, a requirement, or DONE, a description. The location of information within the tree may be underspecified. Linguistic interpretation is modelled as the process of building a semantic tree, and lexical entries provide information on the update of partial tree descriptions. The <LINK> operation allows the construction of more than one tree for the same utterance, providing that some Declarative Unit occurs in both trees. Paradigmatically and historically this is associated with relative clauses, but in fact there is no necessary connection with the relative pronoun, and thus the <LINK> operation is not restricted to cases where there may be lexical

triggering. The claim is rather that it may generally be available as a parsing strategy. This introduces the following questions:

- what constraints are there on the operation of <LINK>?
- does <LINK> have to be licensed in any sense?
- should different types of L-tree be distinguished?
- how does <LINK> affect the overall structure of the utterance?

To answer these questions I look more closely at the nature of <LINK> trees which do not display any lexically realised connection to the initial tree. This first requires an investigation of predication within LDSNL.

Chapter Three

Extending <LINK>:
P-Predication and Reduced Relative Clauses

3.0. Introduction

In the previous chapter I restricted my attention to relative clauses containing a fully tensed verb phrase. I now turn to relative clauses which do not display any kind of tense marking: the reduced relatives in English. These are so-called because of their deficiencies: not only do they lack tense, but they are not introduced by an overt complementiser. I will argue that the <LINK> account described in chapter 2 can be extended to these, maintaining a uniformity of explanation across these phenomena without postulating the need for any kind of zero complementiser. The existence of such 'minimal' relative clauses can shed light on the structure of more complex variants, and the dynamics of their construction. However, in order to develop the account of these reduced relatives, I need to introduce developments to the LDSNL framework. Firstly, I make a distinction between tensed and non-tensed tree structures within LDSNL. I also have to examine more generally the nature of predication within the system and how this is effected by lexical entries across word classes. To these ends, I introduce a new type to the system, type (p), to represent a proposition which is not marked with a location within a temporal flow. I develop lexical entries using this type and relate these to the structural dynamics. I then return to an account of the reduced relatives. These developments will allow a broader swathe of generalisations across what can be termed syntactically dependent clauses, as will be shown in chapters four and five.

3.1. Reduced Relatives

3.1.1. Defining Reduced Relatives

I am using the term 'reduced relative' to refer to relative clauses which do not have an overt complementiser and are non-finite, as shown in (1). This contrasts with (2), what I term a full relative, which does have a complementiser,[1] and which has a finite form of the verb.

(1) The woman [*sitting on the grass*] was eating an ice cream
(2) The woman [*who was* sitting on the grass] ate an ice cream.

It is unsurprising that relative clauses which are not explicitly marked as such are found in natural language. The <LINK> rule in its most general form allows the 'sharing' of a node between a tree and a L(inked)-tree, without the need for any linguistic form to represent that content in the L-tree. Application of the <LINK> rule is a freely available option within the system, so that at any point the task of building an L-tree can be launched. However, it is necessary to distinguish different instances of unmarked relative clauses according to whether or not they are finite, as these display a marked contrast.

In 2.7. I examined *finite* relative clauses in English and noted that, in those cases where these lacked a complementiser, the node which is carried over into the L-tree cannot appear in subject position. This intermediate case, where a relative clause has no complementiser but does contain a tensed verb form, (4b), I refer to as a complementiser-less relative.

(3) a. the woman who came in sat down
    b. *the woman came in sat down
(4) a. the woman who I saw yesterday sat down
    b. the woman I saw yesterday sat down

[1] I use the term 'complementiser' in an informal manner. How a particular 'complementiser' contributes to the structure building task depends on the individual item. In English, for example, 'that' functions differently from 'who'.

In the case of *non-finite* relative clauses with no complementiser precisely the opposite is true - the transferred node has to be in the subject position.[2]

The present focus is on reduced relatives. The process of reduced relative formation in English is extremely rich: all major categories can form reduced relatives.[3] They are characterised by three properties:

i. there is no overt complementiser

ii. there is no tense marking on the clause

iii. the head noun must function as the subject of the reduced clause.

These are illustrated in the contrasts shown in the examples below.

(5) a. the man angry at the management disrupted the meeting.
b. *the man who angry at the management disrupted the meeting.
(6) a. the man angry at the management disrupted the meeting.
b. *the man is angry at the management disrupted the meeting.
(7) a. the man angry at the management disrupted the meeting.
b. * the man John angry at disrupted the meeting.

Note that all of these cases are restrictive relatives.[4] Any <LINK> account of them should be characterised as a process of structure building within the

---

[2] The exceptions to this generalisation are certain cases with the infinitive, such as (i), meaning 'the place where one should be seen' and not 'the place which will be seen'. Similarly (ii), effectively meaning 'the man who should be interviewed' .These are discussed in section 3.6.

(i) the place to be seen
(ii) the man to interview

[3] The status of NP reduced relatives might be considered more controversial. I claim that there are no idiosyncratic properties associated with these and that the ways in which they differ accord exactly with the general problems of NP predication.

[4] There are two important caveats here. Firstly, there exist non-restrictive cases such as (i). These are classified as parentheticals, and are discussed in chapter six.

(i) John, MP for Croydon, voted against the government.

Secondly, there are cases which ought to be non-restrictive but which are in fact clauses restrictive of a proper name.

(ii) John the president is not available. John the secretary will gladly talk to you.

These depend on the possibility of picking out one from a set of individuals denoted by a proper name which is functioning as a predicate.

overall confines of the e-task; the node carried over is predicted to be the metavariable from the common noun.

The reduced relative clause constitutes an environment where English apparently displays pro-drop. The reasons for this will be examined presently, but the explanation will be sought in structural rather than lexical terms. To remove the possibility of external factors coming into play in the formation and licensing of this construction, I consider such constituents only in subject position occurrences, as in (8). What characterises this is, firstly, that the noun phrase plus reduced relative (here a prepositional phrase) together stand as a distinct constituent, and secondly, this complex is external to the main sentential predicate. This is in sharp contrast to (9), where the objects are subcategorised for.

(8) *the letter to Mary* was on the kitchen table.
(9) John gave *the letter to Mary*

In (9), whatever sort of structure is assigned to the object complex, this is in some sense dependent on and incorporated within the verb meaning.[5] Moreover, the two object NPs do not constitute a single unit. The differences between the two can readily be illustrated by their different behaviour with regard to clefting.

(10) it was *the letter to Mary* which was on the kitchen table.
(11) *it was *the letter to Mary* that John gave.
(12) *it was *the letter* which was *to Mary* on the kitchen table.
(13) it was *to Mary* that John gave *the letter*.
(14) it was *the letter* that John gave *to Mary*.

The object complex cannot be clefted together, but the reduced relative, which constitutes a single unit, can. Nor can the reduced relative be broken into its sub-parts for the sake of clefting.[6]

---

[5] For an account of these in the present framework see Marten (in prep.).
[6] In fact it is possible to have discontinuous dependencies in the case of reduced relatives.

(i) *the man* came yesterday *from British Gas*.

This is discussed in chapter five.

In the rest of this section I assume that there is an identifiable class of relative clause constructions, and examine what lexical types can make a well-formed reduced relative in English.

3.1.2. Adjective

Adjectives can occur in reduced relatives; however, single adjectives are dispreferred in this environment: these tend to be realised before the noun, the premodifying position, as shown below, where (16) is the preferred form.

(15) ?the man angry
(16) the angry man

The reduced relative occurs when there is further structure. The examples below demonstrate that it is impossible to place the adjectival phrase in the premodifying position.

(17) a. the man happy with the outcome
b. *the happy with the outcome man
(18) a. the mother proud of her twins
b. *the proud of her twins mother

An obvious area to look for an explanation for this would be in the area of weight, either in terms of putative phonological restrictions, or in terms of constraints on ease of parsing. However, it is not simply a matter of there being additional structure; what matters is what sort of structure this is. Certain additional material is permissible in the premodifying environment. Examples (20) and (21) illustrate that the adjective itself can be modified (in present terms this implies that the type specification of the adjective remains unaltered). (22) shows that the same sort of modifier, ie another adjective, may be used more than once in the same structure; this is a case of the same sort of operation applying again.

(19) the red car
(20) the bright green car
(21) the very red car

(22) the big red shiny car
(23) *the red like the sunrise car
(24) the car red like the sunrise

It is only when additional material is incorporated into the adjective phrase itself, when, semantically, there is a additional material as part of the proposition, that the reduced relative clause is compulsory. Such material has to come after the adjective. When the adjective *combines* with any other elements occurring after it, these can only occur in the position afterwards.

(25) a. *the big as a bus car
b. the car as big as a bus
(26) a. *the happy at the news woman
b. the woman happy at the news
(27) a. *the hungry for knowledge students
b. the students hungry for knowledge
(28) a. *the older than its mate tiger
b. the tiger older than its mate

The requisite notion of complexity is concerned with the nature of combination. The premodification position can only support modification operations; any structure involving predication has to occur as a reduced relative. This corresponds to an analysis of this latter as a <LINK> structure, which is developed in 3.6.

3.1.3. Preposition

Prepositions can occur freely in the post-nominal position as reduced relatives. What concerns me here are those cases where there is no necessary relation between the noun and the prepositional phrase, as in (29), rather than those cases like (30) where it is claimed that the noun phrase apparently requires a prepositional phrase in order to be well formed (cf Grimshaw and Williams 1994).

(29) the man in the kitchen
(30) the destruction of the city

I take it to be definitional of the reduced relative structure that the head noun is self-sufficient in both syntactic and semantic terms. That is, there are no subcategorisation requirements to be fulfilled, and the noun phrase *may* be used to uniquely identify a referent.

In terms of the dynamic approach to structure building being advocated here, it is not the case that all nouns will subcategorise for a number of different expansion possibilities. As Goldberg (1995) argues in justifying a construction-based approach, a theory which assumes discrete sub-categorisation possibilities for each noun phrase in order to allow the possibility of post-modifiers will require potentially enormous lexical entries. In the current framework a number of options are permissible in terms of the structure building operations allowed overall,[7] and at any stage in the derivation process a subset of these will be available. The acceptability of any of the possible subset at a particular point will turn on whether or not the appropriate conceptual structure can be invoked; this is not constrained by the syntax as such.

Both (31) and (32) are perfectly acceptable in terms of syntax; the semantic oddity of (32) relates to world knowledge rather than any information specifically in the linguistic system.

(31) the jug on the table
(32) the moon on the table

Returning to the syntactic properties, prepositions can never occur before the noun phrase in the premodifying environment which is suitable for adjectives.

(33) *the in the mountains house
(34) *the at two o'clock meeting
(35) *the for Jane present

The generalisation drawn for adjectives was that if the adjective combines with other elements in a predicative structure then it has to occur as a reduced relative. This is supported by the behaviour of the preposition which has to

[7] These vary across languages, so for example, English has reduced relatives but not the construct state, whereas Hebrew has both.

combine with a noun phrase following it, and is therefore predicted not to be able to occur as a premodifier.

3.1.4. Nouns

While adjective and preposition phrases can be straightforwardly incorporated into the category of reduced relative, it is not clear that noun phrases are equally uncontroversial. In order to guarantee the restrictive reading, I have up to this point used definite noun phrases. With noun phrases, it seems that these structures are much more problematic.

(36) *the man the president
(37) ?the man a doctor
(38) the man president of the company

(36) is undoubtedly bad, while (37) and (38) are better. When considered in an utterance rather than in isolation, these latter improve considerably.

(39) the man a doctor will be coming in to help with the operation.
(40) the man president of the company has rewarded himself with generous share options.

(37) and (38) are examples of predicational uses of noun phrases, as opposed to the equational example in (36).

This distinction, and the differences in syntactic behaviour it causes, has been much debated in the literature (cf Doron (1983), Partee (1987), Rapoport (1987), Williams (1990), Heycock and Kroch (1997)). The discussion has centred on how this relates to (properties of) the copula. The source of the distinction is the difference between predicational and referential use of the noun phrase. In the latter case, an equational reading is derived, and a number of explanations can be given for this, depending on the analysis given to the copula. However, in these reduced relatives, it is not possible to attribute the difference to any supposed ambiguity in the copula, as this is not present.

From the present perspective it is to be expected that the predicational reading is more available as this should proceed straightforwardly, whereas

derivation of the equative reading requires further operations. Note, though, that also characteristic of the reduced relative clause is the absence of any lexically instantiated material to derive the subject position. It is in precisely this environment that languages as diverse as Hebrew (Doron (1986), Rapoport (1987)), Hausa (Green (1997)), and Irish (Doherty (1996)), generally require some form of copular element to be inserted to obtain the equative structure.[8] Given the absence of such an element in English, where 'be' acts as a verb and is much wider ranging in its function, it is unsurprising that the equative reading is difficult to derive.

3.1.5. Verbs

In reduced relative clauses, non-finite verb forms can be used but finite verbs cannot.

(41) the man eating the sandwich sat down.
(42) the boxer hit full square on the jaw fell over.
(43) *the man ate the sandwich sat down.
(44) the man who ate the sandwich sat down.

This accords with the general characterisation which will be developed of reduced relatives as dependent on other syntactic material for their interpretation, and therefore lacking the tense information associated with main clauses.

3.1.5.1. Present Participle

The present participle, indicating continuous aspect, is well formed as a reduced relative, as the examples below indicate.

(45) the man smiling at the sailor
(46) the woman making an axe
(47) the dog lapping up the milk

[8] I use the term 'copular pronoun' purely as a descriptive term. For the dynamic function of the copular pronoun in Hebrew, see Swinburne (1998).

In this particular environment, I assume that this is the participle form and not a gerund, as there is no reason to suggest any process of nominalisation has taken place. Nor is there any reason to suppose that the participle should *itself* supply any kind of pro/PRO element, which would then be co-indexed with the head noun. The minimal hypothesis is that the '-ing' morphology associated with the participle provides only aspect information. In this context the participle still assigns a theta role which has to pick up external reference. This contrasts with the case where the participle is used 'nominatively', where either the theta role is suppressed or a generic PRO element is supplied, as in example (48).

(48) eating is fun.

The tense of the main clause does not have any bearing on the acceptability of the participle in these reduced relatives:

(49) the man smiling at the sailor walked down the road.
(50) the man smiling at the sailor is leaving the bar.
(51) the man smiling at the sailor will go to the port.

What is interesting is the nature of the temporal dependencies. It is a moot point whether the 'present' participle should be construed as such here or whether it only invokes continuous aspect, and then how tense and aspect interrelate. (49) seems to have available the two readings given in (52).

(52) a. the man who is now smiling at the sailor and who is identified by this description walked down the road at some point in the past
b. there is some event in the past where there was a man who at that time was smiling at a sailor and who walked down the road

Regardless of the overall tense marked, the participle can be construed either as being independent in temporal reference or as dependent on the main proposition. This is further evidence that these reduced relatives should be treated as distinct units. I consider below whether this is a general property of reduced relatives.

With certain verbs, it is possible to place the participle before the noun in a pre-modifying position, as was the case with adjectives. (53) gives an example of this. While it is always possible that the present participle can occur *after* the noun, the circumstances in which it can appear *before* the noun are much more restricted, so (54a) is better than (54b). This contrasts with the behaviour of adjectives, where this is a free process. The participle forms which are acceptable before the noun have in some sense been 'adjectivised', but this process is restricted semantically. What this might mean is examined in 3.5. Additional structure in the participle phrase prevents this option in (55); it is therefore impossible to have a transitive verb in the premodifying position, as shown in (56). (57) shows that this is also the case when there is no lexically realised object.

(53) the smiling man
(54) a. the man running
b. ?the running man
(55) *the smiling at the bus driver man
(56) *the eating bagels man
(57) *the eating man

In cases where both options are available, the positioning of the participle may cause a marked difference in interpretation. In (58) the preferred interpretation is that there is smoke coming off the man's body, for example in a case of spontaneous combustion. (59) can only have the interpretation that the man is smoking something, for example a cigarette.

(58) the smoking man
(59) the man smoking

In the pre-modifying position, only an intransitive reading is available. In the reduced relative, it is not only possible but preferred to infer a (possibly abstract) object. This is to be expected, and provides further evidence for the distinction between a modification operation as such, which occurs before the

noun phrase, and operations after the noun phrases where predication processes are allowed.

3.1.5.2. Past Participle

English morphology complicates the issue of distinguishing between the past and passive participles, and the past form of the verb which, because it is a tensed form, is unacceptable in this environment. Where a past participle is unambiguously such, it is acceptable as a reduced relative, as in (60). Otherwise it will be parsed as a main clause and not assigned a <LINK> structure, as in (61) and (62).

(60) the man gone from the flat
(61) *the man went from the flat
(62) the man disappeared three days ago

These phenomena are subject to dialectal variation in English. Speakers of British English tend to regard (63) and (64) as 'non-standard' but acceptable.[9]

(63) the man sat in the corner
(64) the woman stood by the bar

Acceptability of these as reduced relatives correlates with their acceptability as main clause predicates in the form 'be' plus participle.

(65) the man was sat in the corner
(66) the woman was stood by the bar

Where these are judged acceptable, the corresponding reduced relative is also accepted.

3.1.5.3. Passive Participle

The passive participle is fine in this environment, as the examples below show.

[9] That is to say, these speakers would both produce and process them unproblematically in spoken language but would not write them.

As expected, it is possible to build up additional structure in the reduced relative clause.

(67) the man hit on the head
(68) the clothes bought yesterday
(69) the car cleaned by the boy
(70) the ill health complained of by those who fought in the Gulf

Adjectivisation is also possible for the passive on its own.

(71) the injured man
(72) *the injured by the car man

This corresponds to the patterning displayed by the other participle forms.

3.1.5.4. The Infinitive

The use of the infinitive in reduced relative clauses is also subject to dialectal variation, with a much greater degree of acceptability in British than in American English.

(73) the man to do the dishes
(74) the man to fix the car
(75) the place to be seen
(76) the man to interview

For (73) and (74) there is a correlation between the acceptability of these as reduced relatives and their acceptability as main clauses with the copula providing tense.

(77) the man is to do the dishes
(78) the man was to fix the car

The interpretation for (73) and (74) may be paraphrased as below:

(73)' the man who will do the dishes
(74)' the man who will fix the car

(75), a passive form, is ambiguous between, loosely phrased:

(75)' a. the place which is to be seen [it]
b. the place where one should be seen [there]

Note that the interpretation given in (75'b) requires the instantation of some generic PRO form to be identified from context;[10] this is also the case for (76), which has the interpretation in (76').

(76') the man for us/you to interview is here

Only the specification of such a PRO form blocks the head from straightforwardly acting as the subject of these reduced relatives.[11]

3.1.6. Reduced Relatives and the Copula

I have introduced this data in order to illustrate what may form acceptable constituents in a reduced relative clause. What is striking about the above is that these are all lexical categories that are required to have copula 'be' support in order to be a well-formed main clause in English, ie in cases where they form the main predicate. This is shown in (77)-(83).

(77) the man is angry at the decision
(78) the man is in the kitchen
(79) the man is president of the company
(80) the man is smiling at the sailor
(81) the man is gone from the flat
(82) the man is hit on the head
(83) the man is to do the dishes

---

[10] Sag (1997) proposes a similar account in HPSG.

[11] Note the behaviour of these two examples with 'be' supplying tense:

(i) the place is to be seen
(ii) *the man is to interview

Here PRO cannot be the subject, as in a tensed main clause in English the subject must be supplied by a lexical noun phrase.

In the Principles and Parameters approach, these would be assigned a small clause structure, where the copula selects a small clause form.[12] However, not all of these can appear in typical small clause environments, as shown in (84).

(84) *I consider the man smiling at the sailor.

This can be rescued by the insertion of the infinitive form of the copula.

(85) I consider the man to be smiling at the sailor.

However, the small clause is not a well-defined constituent in the present system.

The data above suggest that in Labelled Deductive Systems for Natural Language parallel structure building operations should be defined for copular constructions and reduced relative clauses. These differ in that the former case requires a lexically realised subject and is explicitly marked for tense; the latter has neither of these properties and is, by hypothesis, a <LINK> structure.

3.2. Reduced relatives and <LINK> (i)

I now turn to consider how to derive reduced relatives as <LINK> structures. (86) gives the rule for the <LINK> operation without a relativiser, repeated from chapter two. This states that from any location annotated with type (e) it is possible to launch the task of constructing an L-tree; the content of the node is carried over, and this content is underlocated in the new tree.

(86) <LINK> Rule Without Relativiser

$$\frac{[_m \ldots\ldots\ldots [_n Fo(\alpha), Ty(e)] .. \bullet\ Ty(t)]}{[_m \ldots\ldots\ldots [_n Fo(\alpha), Ty(e)] .. \bullet\ Ty(t)], [_L \bullet\ Ty(t);\ [_* Fo(\alpha), Ty(e))]]}$$

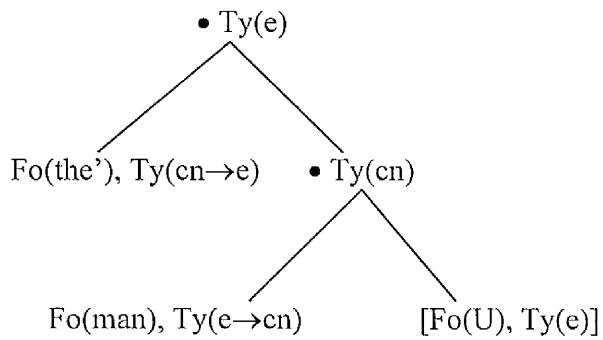
[12] For an overview see the discussion in Heycock and Kroch (1997).

Applying this rule would seem to allow the correct derivation of these reduced relatives, providing that the correct type specification is assigned to each particular lexical class. For ease of exposition I will consider example (87) using a verbal participle, which can uncontroversially be assumed to be of type (e→t).

(87) the man drinking Campari

The derivation for the determiner phrase proceeds standardly, as in (88); the determiner requires a common noun; the introduction of a common noun makes available a metavariable formula (U) which is of type (e). After application of the <LINK> rule this is carried over into an L-tree, shown in (89).

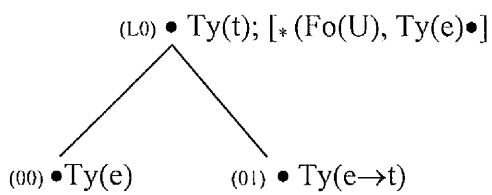
(88)



(89) (L0) • Ty(t); [$_*$ (Fo(U), Ty(e)•]

At the root node of the L-tree, then, there is the requirement for type (t). By the rules of Introduction and Prediction this can be rewritten as the form given in (90).

(90)



The next lexical item to be scanned is the participle. It appears that this can be used, because there is a requirement for type (e→t) at location (01).

pointer to the location where this requirement holds, so that the lexical rule can apply.

The only solution would be to use the underlocated node first, putting it in the subject position (00). As a general strategy this seems stipulatory. Note the contrast to the case where there is a full lexical noun phrase. In that case, if the pointer moves down to the subject node this provides the correct environment to trigger the lexical action, and so the noun phrase will be used to fulfil the requirement.[14] In this case of an underlocated node, however, there is no lexical form and hence no process of scanning to give access to such a lexical rule. There is no reason why the underlocation should automatically be resolved here. A more plausible strategy is to define a new <LINK> rule for these reduced relatives, which could then capture the fact that the linked node *has* to be used as subject of the reduced clause. This has the disadvantage of multiplying the number of <LINK> rules[15] but would seem justified to capture the differences between reduced and other relatives, unless there are some other general process going on which can be identified. In tensed complementiser-less relatives the linked node *cannot* appear as the subject. In the non-tensed case it is not possible for the relative pronoun to be used. There is nothing in the account that derives this unless separate <LINK> rules are to be defined.

A further problem is that as it stands, the system has no way to distinguish between tensed and non-tensed relative clauses. The reduced relatives constitute tenseless clauses; in semantic terms these are propositions apparently lacking any temporal location. Such an object needs to be defined within this system, and this I do in 3.4. First I consider the interaction with tense and the copula. Finite verbs supply tense; those predicates typical of reduced relatives lack tense and hence require the copula to provide tense in order to form a main clause. But does the copula play any further role? I am adopting an analysis where the lexical categories which constitute good predicates in the reduced relatives

[14] That is for English. Case information in other languages would supply the locational information.

[15] cf Kempson, Meyer-Viol & Gabbay (in prep.), chapter three for a discussion of <LINK> rule variations.

function more generally as predicates, that is, as the semantic functor. To pursue this, it is necessary to examine the behaviour of the copula.

3.3. Aspects of the Copula

The copula 'be' in English appears to be associated with a number of possible meanings. It is used in cases of predication, equatives[16] and existentials.

(93) the cat is big.
(94) rats are rodents.
(95) in the kitchen is a dog.

The minimal hypothesis is that 'be' only supplies tense information. This raises the question of how to integrate 'be' into the process of structure building, and the contribution that it makes. The tree structure configurations outlined in this framework indicate semantic relations; there are no functional projections, and hence no distinction is possible between functional and lexical projections. [17] This means that unavailable to this framework is the widely adopted approach of having 'be' associated with a functional projection, combining with a small clause, according to a variety of movement operations.

3.3.1. Copula as Procedural Predicate

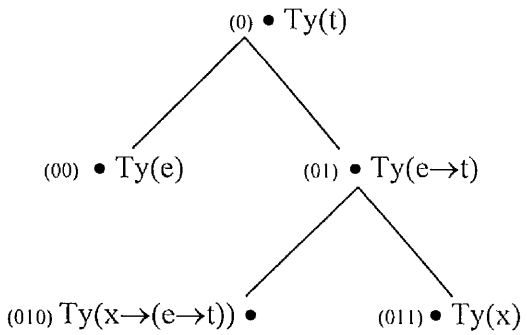
The copula might be analysed in a number of ways in the present approach. It would be possible to have the copula as a purely procedural device. On this analysis, it would just provide a predication frame, but such a frame would not be associated with any conceptual content, ie no information would be supplied for the formula label. That is to say it would drive the process forward in structural terms, supplying the requirements for a subject and some form of

[16] The distinction between predicational and equative uses of the copula diverges from mainstream approaches. The distinction made here is in terms of the type specification and how this operates. The system licenses a default equative but this is associated with a specific structural configuration rather than any lexical material. For the predicative/equational distinction and its syntactic ramifications see the discussions in Rapoport (1987) and Heycock and Kroch 1997).

[17] Tense could be characterised as type (t) → (t), adding a label, though this is not the approach I adopt below. See the detailed discussion in Perrett (in prep.).

predicate, as shown in (96), where 'be' would have the type specification (x→(e→t)).

(96)



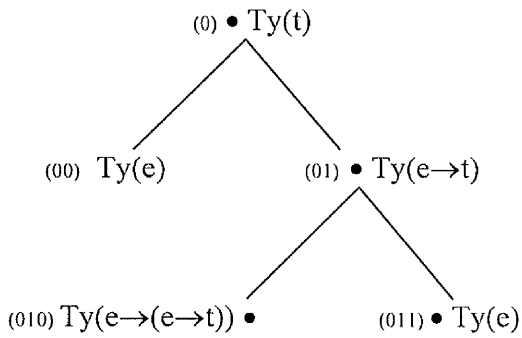
Such an approach could then be manipulated so that 'be' effectively provided the requisite type-shifting operations to guarantee that differently typed lexical items could be used in this structure. The danger here is that in the present approach it is not possible to separate clearly syntactic and semantic operations. That is to say, a certain syntactic structure is derived which then maps onto a semantic representation by means of correspondence rules. The structure as built up has to transparently indicate those relations.

3.3.2. Semantic Considerations

The tree building operation is straightforward, and can be defined in terms of underspecification, which is an important underpinning of this approach. But matters are not that straightforward, in that this would then require separate operations for different categories of lexical item. Consider two possible applications of this. (98) shows the type schema which would apply for the noun phrase example (97).

(97) Terry is the film-maker.

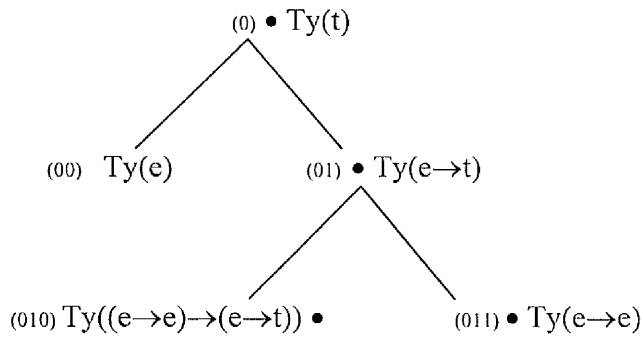
(98)



(100) shows the type schema for an adjective case like (99), assuming adjectives to be typed as noun phrase modifiers.

(99) John is big.

(100)



Here it seems not so much that an underspecified approach is developed as that any notion of the role of the copula in structure building is lost.[18] Specifically, these two trees represent different modes of semantic combination, and yet in terms of structural tree relations are identical.

Note that in the above examples it is the copula itself which is filling the predication node at location (010). This runs contrary to basic intuitions about the semantic relations established. The theta roles and subcategorisation requirements are still coming from the predicate itself.

(101) the wall was red.
(102) the wall is red.
(103) the wall will be red.

[18] In fact the situation is worse, as adjectives can be common noun modifiers.

In the examples above, the property 'red' is predicated of the entity 'the wall'. Nothing about this predication changes except for its location in the temporal flow.

(104) the stone is alive.
(105) John is a beast.

(104) and (105) require metaphorical interpretation. This is connected to the conceptual properties of the predicate and the subject; the copula has no bearing on this whatsoever. It is clear that if the copula is to assign no conceptual content, then nor should it build any semantic structure.

3.3.3. Syntactic Considerations

The syntactic evidence confirms the view that the copula is not the predicate. All the standard syntactic tests to identify the predicate suggest that it cannot be the case that the copula is the predicate. I give representative examples below for each of the categories I am considering.

(106)-(111) are examples of predicate fronting: (106) shows the basic pattern with the verb; the rest of the examples illustrate that for all the cases under consideration here, the copula is never fronted as part of the predicate. (107b) shows explicitly that fronting the copula is ungrammatical; the same holds for all of these cases.

(106) Maria said she would sing the song and sing the song she did.

Adjective phrase:

(107) a. Toby said he was happy and happy he was.
b. *Toby said he was happy and was happy he.

Prepositional phrase:

(108) Toby said he was in the kitchen, and in the kitchen he was.

Noun phrase:

(109) Toby said he was president of the company and president of the company he was.

Participle:

(110) Joan said she was smiling and smiling she was.
(111) Joan said Mary was hit on the head with a rolling pin and hit on the head she was.

This is true regardless of tense, as (112)-(114) show.

(112) Trevor says he was in the park and in the park he was.
(113) Trevor says he is in the park and in the park he is.
(114) Trevor says he will be in the park and in the park he will be.

The same results are obtained with the placement of the sentential adverbial. The unmarked position for this is before the predicate which it modifies, as shown with a finite verb in (115).

(115) Sarah definitely ate the cake.

In the case of the copula constructions, the adverb comes after 'be', indicating that it is the constituent following which is the predicate.

(116) Sarah is definitely in the kitchen.
(117) Sarah is definitely happy.
(118) Sarah is definitely sitting outside.
(119) Sarah is definitely president of the company.

Note that these tests also give the same results where both 'arguments' are definite noun phrases, potentially the most problematic cases for the present approach.[19]

(120) John said Ryan was the champion, and the champion he was.

[19] Application of a default equative rule is invoked.

(121) Ryan was evidently the champion.

This all suggests that in fact 'be' is not supplying the predicate: in structural terms as presently conceived, it does not supply the content of what I have termed the predicate nodes, either location (01) with the requirement type (e→t), or location (010) with requirement (e→(e→t)).

I adopt the analysis that 'be' only supplies tense, and that all other operations have independent sources in the system. This accords with the evidence provided by reduced relatives that in terms of semantic relations contracted, all the lexical categories in (122) are able to function as predicates in and of themselves without the requirement of support from the copula. I will use the term 'basic predicates' to describe these.

(122) Basic Predicates:
Adjective phrase
Prepositional phrase
Noun phrase
Verb participial phrases

Basic predicates have to be able to build predicate structures dynamically. This raises two questions. Is this actually an inherent property they have as lexical items or is it the instantiation of a more general structure building operation? How are tense and predication to be separated in the system? I start with the latter.

3.4 Predication without Tense

I have suggested that it is necessary to define a concept of trees which are not marked for tense. This will lead to questions of what licenses such trees to occur, ie: within what contexts are they well-formed? The answers to these lie in a wider notion of well-formedness for an utterance which I will set aside for the moment.[20]

[20] This is to be addressed in terms of <LINK>.

As outlined above, the overall goal of the system is to derive a tree with root node annotated with a description of type (t). This reflects the fact that the goal of utterance interpretation is to derive a truth evaluable proposition.[21] In order to be truth evaluable a proposition has to be marked for tense. In splitting up the task into its constituent sub-parts, there is the task of deriving a proposition and the task of marking this with tense. This much is not innovative; existing treatments of tense within Labelled Deductive Systems for Natural Language treat tense as a temporal variable acting as a label to the proposition (see the discussion in Perrett (in prep.)). What I propose is to change the terminology and introduce non-tensed propositions as well-formed constituents of the structure building process. As the type (t) is associated with being truth evaluable, I retain this to indicate a fully tensed proposition. The type (p) I use to indicate a proposition which is not marked with tense.[22, 23]

3.4.1. Tree Construction

How does this impact on the process of tree construction? The ultimate goal remains a node annotated with type (t), and this remains the starting point of the operation.

(123) • Ty (t)

Below this is the requirement for a simple proposition and some form of tense marking in order to fulfil the overall goal of truth evaluability. The most straightforward way to implement this in the tree language as presently set up is using the Rule of Introduction. Tense marking is then defined as an operation mapping a proposition onto a truth value. This instantiation of Introduction is given in (124).

---

[21] More accurately in fact, this is a sub-goal associated with linguistic processing, but it is this which concerns me here. The goal of utterance interpretation is to derive contextual effects (cf Sperber & Wilson (1986/95)).

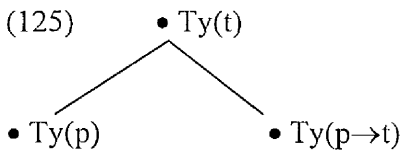
[22] In chapter six this is revised to a distinction in terms of temporal location; at present I am concerned with the way the presence or absence of lexical tense marking affects the type of proposition derived.

[23] The type (t) versus type (p) distinction does not necessarily correspond to a finite/non-finite difference, as will be seen.

(124) $$\frac{[_n \bullet Ty(t)]}{[_n \bullet [_{<d>} Ty(p)], [_{<d>} Ty(p \rightarrow t)]]}$$

Prediction then allows nodes to be built annotated with these requirements. This gives the structure in (125).

(125)

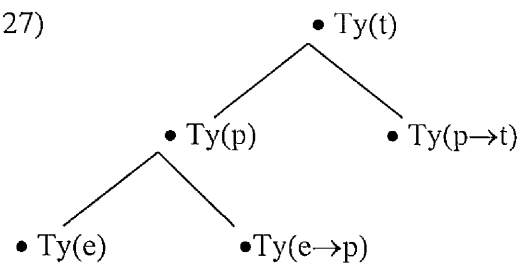


A different instantiation of Introduction can now apply to break down the p-task into its sub-goals, the requirement of an entity and a predicate.

(126) $$\frac{[_n \bullet Ty(p)]}{[_n \bullet [_{<d>} Ty(e)], [_{<d>} Ty(e \rightarrow p)]]}$$

Again prediction can apply to build these nodes, annotated with the requirements. The tree built will be (127).

(127)



This is the standard operation for the development of a main clause. The developments here do not have any bearing on the general course of the derivation as outlined in chapter two. This proceeds as before, except with the difference that the local goal within the proposition is to derive a node annotated with type (p).

Tense information, supplied by the verb, will duly be annotated on the node with requirement ($p \rightarrow t$), at location (01). This is an operation where the tree relations do not reflect steps of function application. There are two reasons to adopt this approach: it maintains simplicity in terms of tree construction, and it reflects the fact that tense information does induce a type changing operation. As discussed with reference to determiners, specific operations can be defined within the system which are associated with a particular structural configuration. I do not go into details of the semantics of tense here,[24] but I envisage an operation which would implement the existing account of tense. This would give the characterisation of the temporal indexing information supplied (this typically underspecifies the value of the time variable derived), and add the temporal variable as a label to the proposition (type (p)) to give the overall formula which is of (type(t)). This latter can then be recast as (128).[25]

(128) $Fo(A') = Fo(S_i : A)$

where A' is of type (t), S_i is the temporal variable and A is of type (p).

However the specifics of this are not of concern here.

The alternative approach would be to keep the temporal labelling process distinct from node building operations. In that case there would effectively be two requirements at the same root node, as shown in (129).

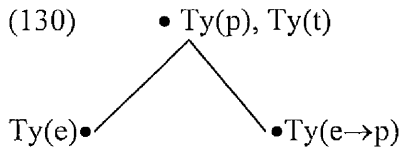
(129) • Ty(p), Ty(t)

The requirement of type (p) would still form the subgoal; the requirement of type (t) would be satisfied once the labelling procedure had been undergone. Tree development would proceed as in (130).

[24] See Finger & Gabbay (1994), Gabbay (1996), for approaches to temporal location in Labelled Deductive Systems and Perrett (in prep.), Kempson, Meyer-Viol & Gabbay (in prep.), for analyses of tense in natural language.

[25] Cf Kempson, Meyer-Viol & Gabbay (in prep.) chapter two.

(130)



The procedures of labelling, and the limits on what sort of information may constitute a label, are at present unresolved in the theory.[26] It is unclear what the ramifications of having two requirements holding simultaneously at the same node would be, indeed whether this can be defined at all. For present purposes what is important is that propositions of type (p), and hence the requirement of type (p) can be set out in terms of the structure; the details of combination at the root node are of no great consequence. For these reasons, I will adopt the specification represented by (124) and (125), where tense supplies information of type (p→t), which also allows greater clarity of presentation.

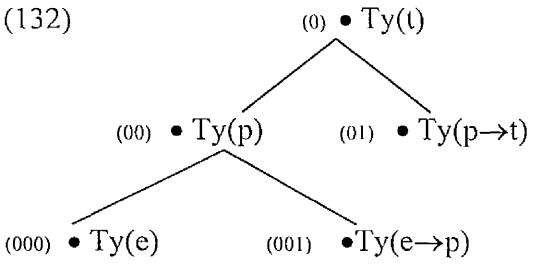
3.4.2. P-Predication

The approach I am proposing maintains the requirement for tense marking on a main clause. Thus in English no well formed representation can be derived from the input in (131). The ungrammaticality of this derives from the lack of tense specification.

(131) John big.

(132) shows the state of tree development immediately prior to the scanning of 'John'.

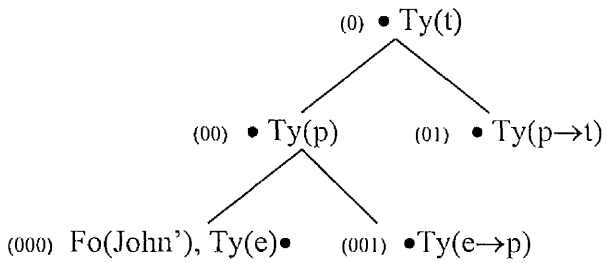
(132)



[26] Tense is a label, as is the Category specification (Q), indicating an interrogative. Whether lexical items which contribute to higher-level explicatures should also constitute labels is a matter for further research.

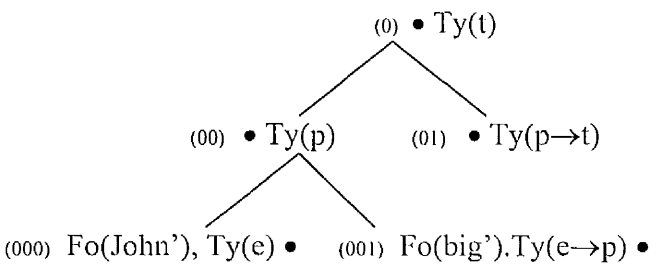
'John' can duly be used to satisfy the requirement of type (e) at node (000), giving the result in (133).[27]

(133)



The next lexical input is the adjective 'big'; giving this the type specification of a predicate,[28] it can be used to fulfil the requirement at node (001), as shown in (134).

(134)



This means that the local sub-goal can be satisfied. However, in the absence of any tense information to locate the proposition in a temporal flow, the proposition remains ill-formed. In terms of the tree, the requirements at node (01) and at (0) remain unsatisfied. A tree with requirements still to do is ill-formed, and hence no well-formed structure can be assigned for the string (the derivation crashes).

What specification should be given to the verb? This approach entirely separates tense information from subcategorisation and conceptual information. The latter annotates the tree at the node where there is the requirement (e→p). Whatever tense information is associated with the particular form of a verb is

[27] This assumes pointer movement to node (000). This is addressed in 3.6.
[28] This is discussed in the next section.

used at the node with the requirement of type (p→t). Similarly, tense information supplied by the copula will be annotated at (p→t).[29]

What is type (p)? Syntactically it is non-finite predication. Semantically it is a proposition with no temporal location. In order to be interpreted, some form of temporal location is necessary. This presents the possibility that trees which are derived as trees of type (p) then inherit tense either contextually or from other trees to which they are structurally connected (for example through the <LINK> operation). The resultant tree would then be the same as one where tense had been lexically supplied, though the process of arriving at this representation would be different. The framework does allow cases where different linguistic input will derive the same final representation, the difference being the route taken to get to such input. An example of this is (optional) resumptive pronouns, where either a gap or a resumptive will result in the same final representation. In the case of tense, however, the situation is different - the lexical input will determine whether or not there actually is any tense marked on the proposition. Tense marking cannot be trivially derived. The property of being marked or not with tense has a direct impact on syntactic behaviour and interpretive possibilities.

The working hypothesis I adopt is that p-trees do not have to be marked with tense. Tense only has to be realised on the main tree in an utterance, which will anchor the temporal location of the utterance as a whole. P-trees do have to be licensed by the contracting of some relation with a main tree. This will be addressed below in terms of different types of p-clauses, how this may derive from the process of their construction, and how the structure of the utterance as a whole is constituted.

3.5. The Dynamics of Predication and Modification

Having introduced the p-tree, I will now turn my attention to the other part of the puzzle - how to derive the structure from the different lexical forms which constitute the basic predicate categories outlined above. The current approach

[29] Aspect information, part of the proposition, is realised on the p-node.

does not allow a definition as simple as 'predication = NP + XP'. A type based approach brings its own problems, but also its advantages. The major problem is that of type specification, and how it is that the system interacts with lexical entries to successfully induce a predication structure: the broader problem is the division of labour between the lexicon and the parser.

3.5.1. Problems of Ambiguity

I assume that the contrast between predication and modification is purely a function of the structural configuration and is not an intrinsic property of the lexical categories as such. Where the lexical item can enter into the structure building process is, however, part of the lexical information. Taking adjectives as the paradigmatic example, in English there is no morphological difference between attributive (135) and predicative (136) uses of the adjective.

(135) The big man.
(136) The man is big.

This is in marked contrast to languages such as Korean, where adjectives can be declined as main clause predicates (137) and display tense information (138).

(137) namca-ka k'e-yo
male-nom big-declarative
'the man is big'
(138) namca-ka k'e-ss-eyo
male-nom big-past-declarative
'the man was big'
(139) k'un namca-ka owa-ss-eyo
big-attrib male-nom come-past-declarative
'the big man came'

As well as the obvious word order difference, Korean being verb final, the attributive form has a distinct set of endings (139). These endings provide the requisite information about combinatorial possibilities in the tree structure. The syntactic ambiguity in English will have to be reflected somehow in type ambiguity in LDSNL.

Whatever type of semantic and syntactic assumptions are adopted, the distinction for linguistic objects between modification and predication is clear: modification takes something and returns the same type or form of object, whereas predication fundamentally alters matters by combining with the object under consideration, ascribing some property to it.

Considered abstractly, modifying an entity still leaves an entity, as in (140). Predicating something of an entity will give a truth value, as in (141), that is to say, the entity either has that property or not.

(140)	e	$e \rightarrow e$	e
(141)	e	$e \rightarrow t$	t

This much is definitional. The difference between the predicative function, from entities onto truth values, and the attributive function, from (for example) entities onto entities, has to be reflected in different type specification.

The problem is that in the present system the type descriptions in a sense have to perform a dual function - both to ensure the correct syntactic modes of combination and to guarantee that the correct semantic operations will be derived.[30] The problem is not that type descriptions determine the order of the string, as the dynamic approach to structure building, involving task states and underspecification, overcomes this area of potential inflexibility. Rather, the abandonment of a clear separation between syntactic and semantic components, indeed between syntax and parser, removes the wider option of having a certain syntactic configuration mapping onto a certain semantic operation, where that mapping itself is defined as a distinct operation of the grammar. In LDSNL the correct structure has to be defined as the operation proceeds according to lexical instructions.[31] This has the advantage over phrase structure approaches of locating the constraints of well-formedness on the *interaction* of any particular structural configuration with input, rather than the listing of static rewrite rules

[30] Of course, if there is a total isomorphism between syntax and semantics this will not be problematic. It is well established that natural language is rarely so well behaved.

[31] Though the caveat holds that this is a *schema* for a parser; where ambiguity or underspecification has to be resolved through pragmatic considerations, this system only presents the available options.

for a category. The process may or may not be able to proceed according to the lexical information inputted after scanning, but this is not determined in advance.

In practical terms, the problem is that all lexical and phrasal cases where there is the possibility of either attributive or predicative use will result in type ambiguity.[32] For example, an adjective such as 'big' has to have lexical entries which reflect that it can either modify a noun or be predicated of one.

(142) a big truck
big Ty(cn→cn)
(143) the truck is big
big Ty(e→t)

For the moment I will focus on adjectives as the clearest example, although the problem is a broader one. The easiest way to avoid ambiguity is to assume one specification or the other and then examine the extent to which this can account for the different cases. As alluded to, and rejected above, there is the possibility of relying on the copula to provide information about the requisite structural operations. This would not, however, account for those cases where there is no copula, which form the central concern of this chapter. Alternatively there is the possibility of having the predicational form as the general form and deriving the modifying uses somehow from this. While this sort of approach could be pursued, as a general strategy it is stipulatory. What requires further investigation is the specific nature of modification and predication in the tree building operation.

Before turning to this, however, I note a further point regarding predication and modification. Relative clauses constitute cases both of predication and

[32] I am ignoring here the general problems of type raising considered in semantics since Montague (1974), for example that a property can be predicated of another property. My aim is to explore an incremental type-based grammar formalism where semantic operations can be performed at certain locations, and the history of an object is not necessarily in any sense transparent to later operations. Bracketing indicates particular semantic steps and the ordering of operations takes on much greater significance. The type of an object is relevant at the point of combination with other premises.

modification,[33] where a predication operation within the relative results in a modification of the head. This is equally true of full relatives and of reduced relatives. Predication takes place within the relative clause in terms of the relations contracted between the constituents, assuming that the syntactic requirements of the individual lexical items must be satisfied in the same way as in a main clause; semantically, some additional property is described as holding of some entity. However the overall effect of this on the main utterance is simply to modify the head noun, as in (144) and (145), in much the same way as happens for example with an attributive adjective, e.g. (146).

(144) the car can corner well
the car → the car which Steve will buy
the car which Steve will buy can corner well

(145) the car can corner well
the car → the car bought by Steve
the car bought by Steve can corner well

(146) the car can corner well
the car → the red car
the red car can corner well

The modification relation, however constituted, has no effect on the predicate argument relations established for the main clause.

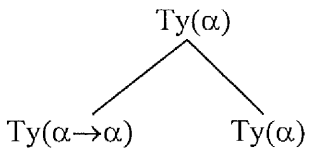
3.5.2. Tree Development for Predication and Modification

The trees derived in this framework reflect the semantic relations established. The primary mode of combination is function application, though as was discussed in section 2.3.2.2. and elsewhere, other operations may be defined to correspond to specific combinatorial configurations. The substantive difference in terms of tree growth is that the modification function does not advance the overall progress toward the goal of the tree. Nor does it change the predicate argument relations (defined in terms of function application). This difference

[33] Such a claim is not special to the current framework. While from a standard syntactic perspective relative clauses may be modifiers, the general approach in the semantic literature is to translate them as (co-ordinated) predicates. This has led to the debate over the interpretation of so-called donkey pronouns. See Lappin & Francez (1994) for an overview of the arguments.

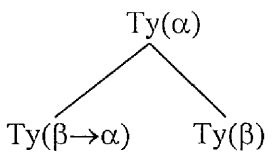
can be readily defined within the model of tree development outlined, as illustrated below.

(147) Tree Configuration for Modification



In modification the overriding goal remains.

(148) Tree Configuration for Predication

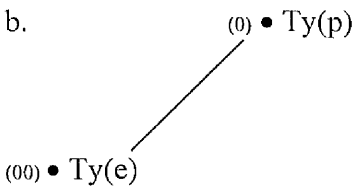


Predication is a self-contained operation, starting with a certain requirement and then fulfilling that requirement in two steps.

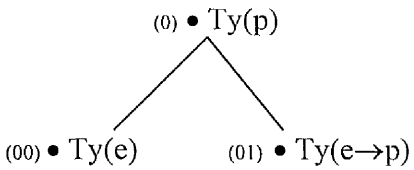
In terms of tree building for linguistic input, what I mean by the predicate is defined semantically, a mapping from an entity onto a proposition. This proceeds according to the tree structure development in (149).

(149) a. (0) • Ty(p)

b.



c.



This is the basic model of predication, where the requirement for a proposition, type (p), can be divided into subtasks according to the rules of Introduction.

Prediction allows these nodes to be built. The scanning of a noun phrase at tree node (00) will fulfil the requirement there. Tree node (01) is annotated with the requirement for some predicate, ie type e→p. For this requirement to be fulfilled requires further lexical input. What can fulfil the requirement at location (01)? This is exactly the environment where the basic predicates can be used: adjectives, prepositions, participles, and (certain) noun phrases. Recall that lexical entries are configured as input/output rules giving the tree update effected by a particular lexical item. The solution is to give all of these items lexical entries where there is a trigger corresponding to the predication requirement.

3.5.3. Context Dependent Lexical Entries

This approach invokes a notion of context dependence where the particular action induced by a lexical item will depend on the state of the parsing process. Stated simply, lexical entries are defined such that the action triggered by a lexical item will depend on the particular environment in the tree. The function of the lexical item is defined according to the triggering context. I extend the system already outlined in that a single lexical item may be associated with more than one triggering environment/update action rule. However, this is not ambiguity in the sense of all possibilities being available at any particular point, and some process of evaluation then being necessary. Rather the process automatically determines the update action on the basis of the trigger. The way the tree develops according to each lexical instruction then determines the triggering environment against which further lexical input is processed. This type of context dependence is an efficient strategy in cognitive and processing terms; rather than having to peruse all the available options at a particular point, the parser pursues one course of action. Lexical items may be ambiguous in terms of making available a number of update procedures, but at the point of structure building there is no ambiguity. I will now consider what the types of rules look like.

3.5.3.1. Lexical Rules for Predicative Context

Lexical entries can be specified in terms of the type (p). If the process of structure building is at a location with the requirement of type (e→p) then the basic predicates can duly build a predicational structure. For example (150) states that in such a location the effect of an adjective is to annotate the node with the formula (big') and the type (e→p).

(150) Adjective
big
'big'
Tr [•Ty(e→p)]
Ac f_{DONE} (Ty(e→p). Fo(big'))

(151) Preposition
in
'in'
Tr [• Ty(e→p)]
Ac $i<d>_0$,
f_{TODO} (Ty(e→(e→p)),
f_{DONE} (Ty(e→(e→p)), Fo(in')),
$\pi<u>$,
$i<d>_1$,
f_{TODO} (Ty(e))

(151) indicates the need for the preposition to combine with another noun phrase which follows.

(152) shows the verbal participle. Note that the trigger will be the same as for a tensed verb - the difference is in the morphological form and the information which this supplies.[34]

[34] There is a separation into the meaning component (formula and type information) and the temporal component (which supplies information to be located elsewhere in the tree). For the breakdown of English verb forms into discrete items according to phonological domains see Kaye (1993).

(152) Verbal Participle
kissing
'kiss'
Tr [• Ty(e→p)]
Ac f_{DONE} (+ cont),[35]
$i<d>_0$,
f_{TODO} (Ty(e→(e→p)),
f_{DONE} (Ty(e→(e→p)), Fo(kiss')),
$\pi<u>$,
$i<d>_1$, f_{TODO} (Ty(e))

(153) shows a schematic entry for a noun phrase.

(153) Noun Phrase - general schema
Tr [• Ty(e→p)]
Ac $i<d>_0$,
f_{TODO} (Ty(e→(e→p)),
f_{DONE} (Ty(e→(e→p)), Fo(?)),
$\pi<u>$,
$i<d>_1$, f_{TODO} (Ty(e))....

This states that a predicating node is built which is conceptually underspecified, and then processing of the noun phrase occurs (ie action standardly invoked by the requirement of type (e)). The content of the noun phrase will determine what sort of relation is inferred as the content of the node of type (e→(e→p)). In the case of two independently referential noun phrases, for example, the equative will be supplied.[36] Complicating the picture is the question of whether or not a noun phrase occurring in the predicative environment is itself being used predicatively or referentially and how this associates with the type system.

In English, type (e) phrases (eg 'a man') can be used predicatively. A number of options are available in specifying actions on a predicative context for determiners and common nouns. (154) presents such a schema where the determiner creates a predicate from the common noun. In this case the formula label at node (e→(e→p) contains the quantificational operator associated with the particular determiner.

[35] ie continuous aspect is marked on the node of type (p).
[36] See Swinburne (1998) for an account of how this works in Hebrew.

(154) Determiner in a Predicative Context
Tr [• (e→p)]
Ac $i\langle d\rangle_0$,
f_{TODO} (Ty(cn→(e→p)),
f_{DONE} (Ty(cn→(e→p)), Fo(α)),
$\pi\langle u\rangle$,
$i\langle d\rangle_1$, f_{TODO} (Ty(cn))

A detailed examination of the possibilities is beyond the scope of the present work, and must await further research.

I allow (154) as a possibility for genuinely predicative cases. Nonetheless, I adopt the general predication frame in (153) for the equational cases which form the main area of concern in the following chapters.

3.5.3.2. Lexical Rules for Pre-Modifiers

In pre-modification environments the local requirement remains, but it is carried down a level (cf the tree illustration in (147)).

(155) Adjective
big
'big'
Tr [• Ty(cn)]
Ac $i\langle d\rangle_0$,
f_{TODO}(Ty(cn→cn),
f_{DONE} (Ty(cn→cn), Fo(big')),
$\pi\langle u\rangle$,
$i\langle d\rangle_1$,
f_{TODO} (Ty(cn))

The effect of this is to build a new node at the left branch, and annotate that with the formula information associated with the concept 'big', move the pointer back up to the starting point, and then build a right branch node labelled with the requirement of type (cn). As regards the semantic operation associated with such a tree configuration I have little to say. Note that this prejudices nothing in terms of how the adjective might relate to the common noun

In the case of modification, this makes no contribution to the overall goal of deriving a node of type (e). This gives an indication of why modifiers occur

before the common noun. If they did not then the process would already be complete. To have a *modifying* operation after the noun would require the restructuring of already established tree relations, which is antithetical to the overall approach adopted. This also suggests the reason for the impossibility of further structure associated with an individual pre-modifying constituent. The building of further structure internal to that constituent is only possible with the correct tree configuration, and this only obtains in a predicative structure.

This schematic approach provides the tree update function associated with pre-modification in English, and so it can be generalised. Those participle forms which have this rule in their lexical entry[37] will be usable as pre-modifiers.

(156) 'Adjectivalised' Verbal Participle
smiling
'smile'
Tr [•Ty (cn)]
Ac $i\langle d\rangle_0$,
f_{TODO}(Ty(cn→cn),
f_{DONE} (Ty(cn→cn).Fo(smile')),
$\pi\langle u\rangle$,
$i\langle d\rangle_1$,
f_{TODO} (Ty(cn))

In this case the morphological marking, combined with the meaning of the concept, allows use as a pre-modifier.[38] What this illustrates is the way in which lexical category becomes more a function of tree development properties rather than an intrinsic property of 'words'. 'Adjective' becomes a categorisation over lexical items according to what operations they induce. Syntactic categories are defined in terms of description of a set of the lexical transition rules. Adjectives, prepositions and nouns can be used as predicates as well as modifiers, while verbs can be used as modifiers as well as predicates.[39] The derivation of types

[37] This being determined by semantic factors beyond the scope of the present discussion. Note however that this does not allow for further combinatorial operations, so any verb requiring an object is intrinsically ruled out.

[38] In this case progressive aspect combined with an intransitive verb to give a stative description.

[39] This is due to the lack of morphological information encoded in English. Where this is not the case, categorial information is associated with the transition rules and generalisations over categories are again metalevel descriptions.

over (lexical) classes, from a dynamic perspective, only makes sense in terms of being some metalevel statement about some particular instance of use of a lexical item. The property of being of a certain type, strictly speaking, holds only of declarative units: ie at a certain location there is a declarative unit having a certain type.

The general problem that lexical items may encode more that one update function according to the particular environment is solved here in terms of context dependent lexical rules.

3.6. Reduced Relatives and <LINK> (ii)

The system now has the means to build non-tensed trees as the structure for reduced relatives. This requires the recasting of the <LINK> rule, as it is defined only for tensed trees, ie those of type (t). The other distinguishing feature of reduced relatives is that the linked node has to be used in subject position. There are two ways to capture this. As there is the need anyway for a separately defined <LINK> rule, this constraint could be stated directly by writing it into the specification of the rule. The alternative is to seek to derive it from more general operations within the system.

3.6.1. <LINK> Rules and the Compulsory Subject Constraint

The first approach is given in the rule in (157).

(157) <LINK> Rule for P-Predication, Compulsory Subject

$$\frac{[_m \ldots\ldots\ldots [_n Fo(\alpha), Ty(e)] .. \bullet\ Ty(t)]}{[_m \ldots\ldots\ldots [_n Fo(\alpha), Ty(e)] .. \bullet\ Ty(t)], [_{L0} \bullet\ Ty(p);\ [\ \langle d \rangle\ Fo(\alpha),\ Ty(e))]]}$$

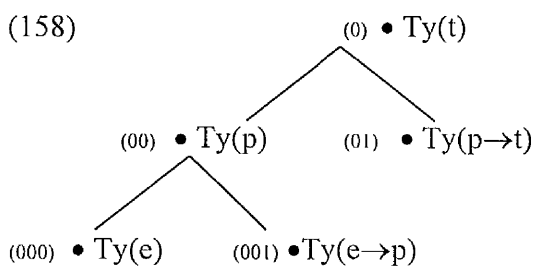
This states that given a location of type (e) in a tree, it is possible to launch a new tree, linked to the existing one, where the goal of that tree is type (p). By definition, the node shared with the main tree, from where the task to build the

L-tree has been launched, has to be used as the subject in the L-tree: that is to say, it is located at a daughter of the root node.

The disadvantage of this approach is that it qualitatively changes the nature of the <LINK> operations that the system makes available. Hitherto, although different characterisations have been proposed according to the specific relative constructions (cf Kempson, Meyer-Viol & Gabbay (in prep.), chapter three), all of these have shared the property that the declarative unit held in common between the two trees is characterised as an underlocated node in the L-tree. This is a defining characteristic of these <LINK> rules. The location of the shared node in the L-tree is subject to the requirements of that tree. According to (157), however, the position is fixed in advance. Moreover, while descriptively adequate, such a formulation is highly stipulatory.

The alternative is to seek explanation elsewhere within the system. Here I return to the problem of the pointer. In 3.2. I discussed why pointer movement presented a problem in deriving L-trees for reduced relatives. In 3.4. this arose as a problem in deriving the p-task. For a main clause the initial steps of tree-building proceed as in (158), repeated from above.

(158)



Where should the pointer be at this stage in the derivation? Should both nodes be built according to Prediction, or should only Introduction occur, and the nodes themselves not yet be built?

Suppose that an NP is now scanned. For the lexical rule for a noun phrase to apply, there must be a trigger with the requirement of type (e), so the pointer has to be at location (000). This can only be advanced by the application of Prediction. Nothing else will induce movement of the pointer. In the parsing of English there is, of course, the freely available option of using the initial noun

phrase in a string as the subject. Indeed this is the default strategy standardly followed. What is required then is a transition rule to reflect the possibility of adopting this strategy, which is available as an option and does not require lexical triggering. This rule, shown in (150), gives a specific instantiation of Prediction for English.

(159) Rule for Non-case Marked Subject Derival
(the Subject Rule)

$$\frac{[_n\ [<d> \bullet Ty(e)]\ [<d> \bullet Ty(e \rightarrow p)]]}{[_{<d>0} \bullet Ty(e)]}$$

This states that if the pointer is at a node such that that node has requirements for two daughters, the left one where there is a requirement of type (e) and the right one where there is a requirement of type (e→p), then the left node can be constructed with the requirement of type (e). This causes the pointer to be moved down to this node. This means that any information entering the process dependent on a type (e) requirement can be used.

(160) gives the equivalent rule which allows the predicate node to be built from a node with the requirements for subject and predicate as daughters.

(160) Rule for Predicate Derival
(the Predicate Rule)

$$\frac{[_n\ [<d> \bullet Ty(e)]\ [<d> \bullet Ty(e \rightarrow p)]]}{[_{<d>1} \bullet Ty(e \rightarrow p)]}$$

This allows the node to be built with the requirement of type (e→p) in those cases where no subject has yet been derived.[40]

When (159) applies and the requirement for type (e) is fulfilled, the rule in (160) does not need to apply. In that case Completion moves the pointer back up to the node with requirement of type (p), and Prediction can then build the node with the requirement of type (e→p). The effect of (159) is to allow that any

[40] The availability of this in English is extremely restricted.

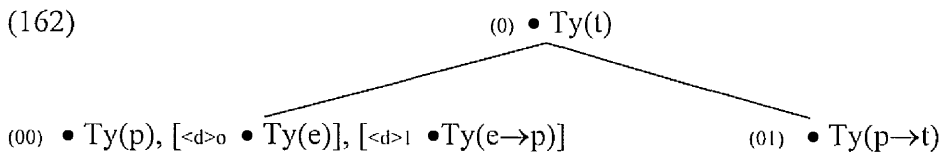
noun phrase encountered at the stage in tree construction before the predicate can be used as the subject. This correctly predicts the strong preference for initial subject in English. Any deviation from this has to be intonationally marked. However, this does not require that the first noun phrase *has* to be used as the subject.[41]

The effect of this in a main clause is to allow the structure building operation to proceed as follows for the example in (161).

(161) Dana eats bananas.

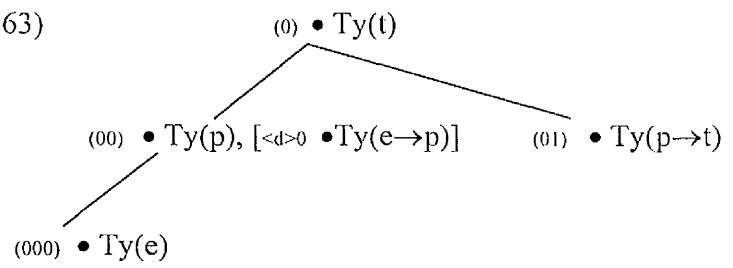
The structure in (162) is derived automatically; the pointer is at node (00).

(162)



Application of the subject rule allows building of a node with requirement of type (e) at location (000), to which the pointer duly moves. This is shown in (163).

(163)



The noun phrase 'Dana' is scanned, supplying the information in (164).

(164) *dana*
'Dana'
Tr [• Ty(e)]
Ac f_{DONE} [Ty(e), Fo(Dana')]

[41] For the three possible strategies available utterance initially in LDSNL see Kempson & Meyer-Viol (1998).

This lexical rule can apply, because there is the correct triggering environment at location (000). Thus, the noun phrase can be used as the subject. At this stage, completion moves the pointer back to (000) and Prediction allows a node to be built at (001) with the requirement of type (e→p). The verb is then scanned and the rest of the structure built.

What impact does this have on the case of reduced relatives? It means that the transferred node, underlocated in the L-tree, can be used as the subject.[42] The effect derives from a rule which is independently required in the system. Now the <LINK> rule defined for p-predication does not require any special characterisation to capture the properties of the underlocated node. This follows from general principles. The rule can now be stated in the more general form below, which does not deviate from the established properties of <LINK> procedures.

(165) <LINK> Rule for P-Predication (revised)

$$\frac{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] .. \bullet \, Ty(t)]}{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] .. \bullet \, Ty(t)], [_L \bullet \, Ty(p); [_* \, Fo(\alpha), Ty(e))]}$$

The difference here from the previously defined <LINK> rule is minimal, the difference being only in terms of the goal specification.

3.6.2. Structure Building for Reduced Relatives

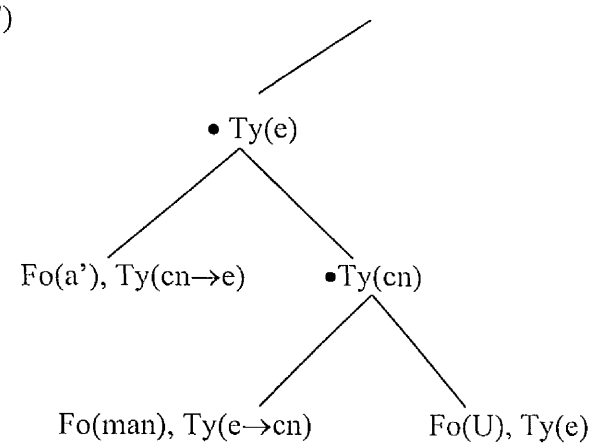
All the pieces are now in place to see how structure is built for reduced relative clauses. I show this with the verbal participle example in (166).

(166) a man drinking Campari ordered another.

[42] For an underlocated node to be incorporated at a certain location, the pointer has to be at that location. This follows from the Merge rule in Kempson, Meyer-Viol & Gabbay (in prep.) chapter two, to which the interested reader is referred. The details of this do not pertain to the present discussion.

The process of structure building for the main tree proceeds as normal. I omit the details of this. In the e-task there is the structure shown in (167) after 'the man' has been scanned.

(167)



At this stage the <LINK> rule given in (165) applies to give the initial specification for the L-tree as in (168).

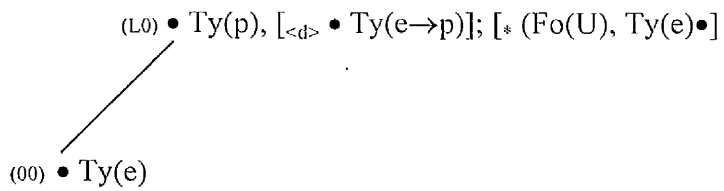
(168) $_{(L0)}$ • Ty(p); [$_*$ (Fo(U), Ty(e)•]

The root node of the L-tree is annotated with the requirement of type (p). The rules of Introduction and Prediction allow this to be rewritten as (169).

(169) $_{(L0)}$ • Ty(p), [$_{<d>}$ • Ty(e)], [$_{<d>}$ • Ty(e→p)]; [$_*$ (Fo(U), Ty(e)•]

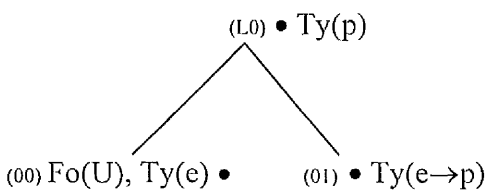
The subject rule, (159), allows the pointer to move to (00), illustrated in (170).

(170)



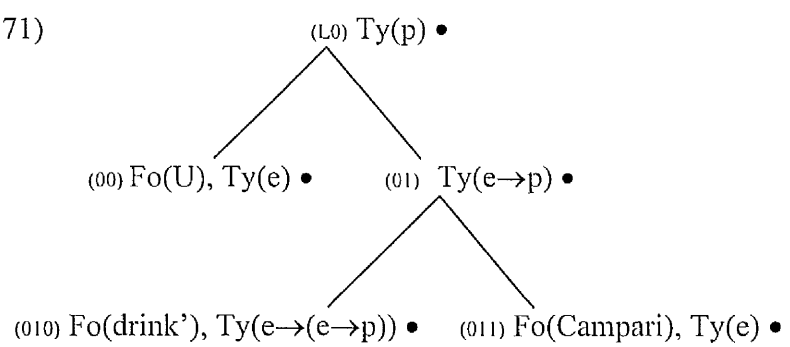
(171) shows the result of the underlocated node being incorporated as the subject. This satisfies the requirement of type (e) at node (00), which consequently allows Completion and Prediction to occur.

(171)



What is now required is that there be a predicate of type (e→p). The next lexical item is the participle. This is scanned: the specified trigger is indeed the requirement of type (e→p), and so the tree update rule supplied by the lexical entry can apply. This in turn requires an object which is supplied by 'Campari'. The overall result is that the underlocation is resolved, all the requirements are fulfilled and all the lexical premises have been used. Compilation can take place and the pointer subsequently returns to the main tree.

(171)



What is derived from the e-task overall is a composite epsilon term combining two co-ordinated predicates, where two properties are predicated of a variable.

(172) (ε, x, (man(x) & drink(campari)(x)))

Note that if the underlocation of the shared node is not resolved at node (00), the derivation cannot proceed at all, as there is nothing to induce movement of the pointer. Moreover, there simply is no other requirement in the tree of type (e)

such that the underlocation could be resolved there. This is the reason that in reduced relatives the shared node can only be used as a subject. The only exception to this is when some subject is supplied as an intrinsic part of the predicate, the infinitive cases supplied with a generic PRO. In contrast, in finite relatives the subject must be lexically realised, either by the relativiser, or in complementiser-less relatives by a full noun phrase unconnected to the head.

3.7. Conclusion

In this chapter I have been concerned with extending the <LINK> account of relative clauses to cover reduced relatives. These are formed by what I term basic predicates, lacking tense, but otherwise able to form propositions. The introduction of type (p) allows these to be modelled as predicates which can combine in standard tree building operations. Trees of type (p) give a proposition, but one that is not marked for tense. The main clause equivalents of these require tense, which is supplied by the copula 'be'. Context-dependent lexical rules allow the definition of modifying and predicating uses for the same lexical item. While having more than one update rule available, the choice is dictated by the stage reached in the parsing process and provides a dynamic alternative to characterisations in terms of ambiguity. The introduction of this new type specification allows a straightforward account of reduced relatives using the <LINK> operation. This differs from that for tensed relatives only in terms of the type specification of the tree derived.

<LINK> allows the structure building process to construct a new tree. This tree must contain an occurrence of the declarative unit which annotates the node from which the <LINK> operation was launched. In the following two chapters, I examine how the <LINK> analysis can be developed to cover cases where the location in the tree at which <LINK> applies is different from the location of the declarative unit information held in common between the trees.

Chapter Four

Discontinuous Constituency and Type (t) <LINK>: Extraposed Relative Clauses

4.0. Introduction

In the last two chapters I have outlined how tree development proceeds for relative clauses in Labelled Deductive Systems for Natural Language. The theoretical apparatus that has been set up provides a uniform *operation* for the derivation of full and reduced relatives. These are assigned *structure* as L-trees, which may be of type (t) or type (p), respectively. All the cases discussed above involved uninterrupted linear continuity between the noun phrase modified and the relative clause, where the relative and its linked node are adjacent, that is to say, where the noun modified and the relative are not separated by any other material in the input string. I now examine how this account can be extended to cases where the relative and its antecedent are not adjacent, where there is intervening material. The <LINK> approach provides an analysis which cuts across divisions previously made in the literature. These are cases of extraposed relative clauses, extraposed prepositional phrases and circumstantial predication. I argue that all these cases can be given a uniform structural configuration as L-trees, though these may differ in their overall properties and their role in the utterance. The different sorts of L-trees can distinguished according to the following criteria:

i. the presence or absence of a relativiser;
ii. the presence or absence of tense marking;
iii. whether the node linked is the whole of the e-node, or the metavariable associated with the common noun.

The interplay of these factors will lead to differences in interpretation, which are ultimately to be explained within a broader model of utterance structure, one which provides for the incorporation of pragmatic factors. Different patterns of occurrence result from the procedural tasks associated with various semantic configurations; the dynamics of the process play a direct part in this. The <LINK> analysis thus allows the restrictive/non-restrictive distinction to be redefined.

In this chapter I analyse the way in which <LINK> trees of type (t) are built for 'extraposed' relative clauses. The wider claim is that the present framework can provide a unified account of discontinuous constituency. I restrict my attention to tensed relative clauses which map onto L-trees of type (t). In the next chapter I consider type (p) L-trees.

4.1. Extraposition

The <LINK> operation provides a unified analysis for a number of phenomena which have been given distinct analyses in the literature. In this chapter I am concerned with the phenomenon of extraposed relative clauses. The term 'extraposition' stems from a transformational view of syntax where movement operations derive different levels of structure.[1] Extraposition from the noun phrase occurs when a modifying constituent is detached from its head and moved to the right. This does not have any effect on the truth conditional meaning.

The relative clause in (1) is extraposed, "moved rightwards", to give the sentence in (2).

(1) a man who was happy arrived.

[1] I am not concerned here with cases of 'it'-extraposition illustrated in (i) and (ii); this term is from Rosenbaum (1967).

(i) That they are lying is clear.
(ii) It is clear that they are lying.

For an analysis of expletive elements in LDSNL see Kempson, Meyer-Viol & Gabbay (in prep.).

(2) a man arrived who was happy.

In both of the examples above the relative clause restricts the interpretation of the noun phrase, 'a man'. Extraposed prepositional phrases are given the same analysis as extraposed relatives. In these it is similarly the case that the prepositional phrase, taken as an NP modifier, restricts the interpretation (see Guéron 1980).

(3) a man with green eyes arrived.
(4) a man arrived with green eyes.

According to the analysis I have given in chapter three, if these prepositional phrase examples involve extraposition, it should be extraposition of a reduced relative. From this perspective, examples such as (6) might also be analysed as involving an extraposed reduced relative.

(5) a man angry at the decision arrived.
(6) a man arrived angry at the decision.

In this case, however, there is a difference. Extraposing the adjective changes the interpretation: it is not that the adjective here restricts the reference of the NP as in (2) and (4);[2] rather, it is interpreted as a more general modifier, some kind of 'adverbial'. These adjectival cases are referred to as depictives or circumstantials, and have been analysed very differently. They have been assimilated to the general class of secondary predication; the structure proposed is either as sister modifiers or as small clauses (see Rapoport 1991 for an overview).

The other classes of basic predicates discussed in the previous chapter seem to pattern with the adjective, in that they provide modifying material rather than restricting the noun phrase.

(7) a man arrived wearing a track suit.

[2] The standard claim is that the prepositional phrase in (4) is restrictive. This is addressed in chapter five.

(8) the man turned up a shivering wretch.

I discuss all of the basic predicates in chapter five. I examine in particular the apparent differences between extraposition and modification, and how this dichotomy applies to prepositional phrases.

I am using the term 'extraposition' here purely as a matter of descriptive convenience. If the basic assumption is that these constructions involve movement, then it certainly makes sense to describe them as extraposed; however, if they are not associated with movement, which indeed is not a theoretical construct in the framework I am using, then 'extraposition' as a label becomes less meaningful. I retain the term for the present for the sake of consistency with the literature, where it is the standard term for the data under discussion.

In this chapter and the next I also use the broader term 'discontinuity'. By 'discontinuity ' I refer both to extraposition as generally understood and to the cases of secondary predication with basic predicates referred to above.[3] I adopt this term as the most neutral, without reference to any underlying level(s) of representation on which some putative extraposition operation takes place. I am suggesting only that there is some (interpretive) relation contracted between the noun phrase and a predicate which is not contiguous to it in the string. All the cases of relatives, both full and reduced, discussed up to now are described as discontinuous when not immediately adjacent to their antecedent. Specifically, I propose that both sets of data are <LINK> structures. This is not to gloss over the differences between 'restrictive' extraposition and 'adjunct' predication; but these differences will be seen to stem from the nature of the <LINK> operation and how it relates to the e-task, and more general pragmatic considerations.

[3] I restrict this to what have been termed depictives or circumstantials, additional predicates which are associated with the subject. I am not considering resultatives or other predicates associated with the object, as in (i) and (ii).

(i) Eric painted the wall white.
(ii) Tony ate the meat raw.

The current research is concerned with optional predication structures, so I exclude on principle cases where there is the possibility of interference from the semantics of the verb.

Prepositions are the intermediate case where the interpretation derived may be either as a restrictive or as a modifier.[4] These provide evidence, then, that there is nothing lexically determined about the differences in these constructions, but that the interpretive possibilities rest on the dynamics of structure building, which the lexical input may underdetermine. This would be implausible and difficult to capture on accounts where two significantly different structures are proposed. I outline below how extraposed relative clauses exhibit both patterns of behaviour.

4.2 Extraposed Relative Clauses

I start by considering the properties of extraposed relative clauses which contain a relative pronoun and which are marked for tense. There are no absolute structural restrictions on the ways in which the noun and the relative are interrelated. The head noun to which the wh- relative is adjoined may be subject, object or adjunct in the main clause. The position in the relative itself may also be subject, object or adjunct. These variations are illustrated below.

(9) shows an extraposed relative clause the antecedent of which is the subject of the main clause; the relative pronoun is the subject of the relative in (a), the object of the relative in (b), and in an adjunct position in the relative clause in (c).

(9) a. a man ate a hamburger with chips yesterday who said he was vegetarian.
b. a man ate a hamburger with chips yesterday who John likes.
c. a man ate a hamburger with chips yesterday who John goes to classes with.

(10) and (11) illustrate the same patterns for cases where the antecedent is the object of the main clause and an adjunct within the main clause, respectively.

(10) a. a man ate a hamburger with chips yesterday which was very tasty.

[4] In fact, as will become clear, not all cases of extraposition of full relatives are actually restrictive.

b. a man ate a hamburger with chips yesterday which John made.
c. a man ate a hamburger with chips yesterday which John made the bun for.

(11) a. a man ate a hamburger with chips yesterday which were low fat.
b. a man ate a hamburger with chips yesterday which John had fried.
c. a man ate a hamburger with chips yesterday which John made a sandwich with.

These may be subject to considerations of plausibility in terms of the ease with which a certain interpretation can be derived. For example, (10c) may be difficult in a culture with no tradition of making chip sandwiches. Nonetheless, there are no grammatical restrictions to rule these out.

In terms of the <LINK> analysis of relative clauses, these data are unsurprising. If the <LINK> operation is a freely available strategy, there can be no motivation for a requirement of strict adjacency between a relative and its head; if the relative pronoun is an anaphoric device it should initiate a search procedure for a suitable referent, but not necessarily the closest; and the relative pronoun guarantees that the shared node is incorporated into the <LINK> tree but does not specify the exact location. The details of this are addressed in 4.6.

4.2.1. General Syntactic Properties

In this section I set out the empirical properties of extraposed relatives identified in the literature as requiring explanation; these present problems which cut across neat distinctions between syntax, semantics and pragmatics. The subtleties they manifest are not well suited to analysis along purely structural dimensions. I do not give here a detailed consideration of all the debate in the literature over the possibilities of structural description, given that much of the argumentation advanced has essentially been linked to particular paradigms, bound to general tools of description which may since have been abandoned for independent reasons. The transformational generative literature has been concerned with whether or not extraposition is a result of movement, and what the attachment site of the extraposed relative should be. Much of the debate has revolved around theory-internal issues about the correct mechanisms for the derivation of grammatical constraints, and I do not review those here (for

example the proposal in Chomsky (1973) to reduce all constraints on movement, both leftward and rightward, to Subjacency; Baltin's justification for an analysis based on binding under government rather than bounding (Baltin 1981, 1983, 1984)). Such notions cannot be articulated and have no meaning in the present approach.

4.2.1.1. Clause Boundedness

From Ross (1967) onwards it has been argued that leftward movement and rightward movement are dissimilar; that is, they display such different properties they are unlikely to be realisations of the same operation. Movement to the left, e.g. wh- movement, is relatively unconstrained, as shown in (12), while movement to the right is clause bounded, as shown in (13) (the example is from Kroch and Joshi 1987:129, their 52). This is Ross's Right Roof Constraint, which states that an extraposed constituent cannot be moved out of the sentence in which it originates.

(12) who do you think that Mary said that Peter wants to know if John loves _ or not?
(13) a. that someone would come who could help became certain.
b. that someone would come became certain that would help.

In terms of Labelled Deductive Systems for Natural Language, this has to be accounted for in the formulation of (restrictions on) dependency relations between trees and the tasks that characterise them. This is discussed below.

4.2.1.2. The Debate over Constituent Structure

There is variation as to where the relative is said to be adjoined. It has generally been claimed that in extraposition from the object, the relative has to be attached within the verb phrase. This is proposed on the basis of standard constituent structure diagnostics such as verb fronting and VP ellipsis (see *inter alia* Baltin op.cit., Guéron and May 1984, Culicover and Rochemont 1990, Rochemont and Culicover 1990). Baltin notes that in detachment from within a subject NP,[5] the

[5] 'Detachment' is the term Baltin uses for his rule of rightward movement.

movement is to a position outside the VP, whereas in cases of detachment from an object NP, the relative cannot be moved outside the VP. This is illustrated below: (14) gives the basic extraposed structure. In (15) the extraposed relative cannot be attached to the verb phrase, as when this is fronted, the result is ungrammatical. In (16), when the extraposed element is associated with the object, the verb phase containing this can be fronted unproblematically.

(14) a man arrived who hailed from Westmoreland.
(15) *Simon said that a man would arrive who hailed from Westmoreland and arrive who hailed from Westmoreland a man did.
(16) Simon said that he would meet a man at the party who hailed from Westmoreland and meet a man at the party who hailed from Westmoreland he did.

While most authors have agreed that subject extraposition has to be outside the verb phrase, for example attached to S' or I', Rochemont and Culicover (1990) cite examples such as (17) and (18) as evidence that this cannot *always* be the correct configuration (their 15 and 16).[6]

(17) a MAN came in with blond hair, and a WOMAN did [e] TOO.
(18) although none of the MEN did, several of the WOMEN went to the concert who were visiting from Boston.[7]

These suggest that in specific environments, phrases extraposed from subject relatives *may* be attached within the verb phrase.

Of these examples, (17) seems better to me than (18). In chapter five I argue that prepositional phrases differ in terms of extraposition from full relative

[6] The examples below (Rochemont and Culicover's (17) and (18)) show that VP ellipsis does not *have* to carry over the extraposed relative.

(i) A MAN came in with BLOND hair, and a WOMAN did [e] with BROWN hair.
(ii) Although none of the MEN did who were visiting from NEW YORK, several of the women went to the concert who were visiting from BOSTON.

This can be analysed either as being such that here the extraposed relative is attached to IP, or that it is still attached to VP but that only part of the VP is carried over, as in (iii).

(iii) John fed the cats in the kitchen and Steven did in the living room.

[7] Where capitals indicate stress.

clauses in such a way that the above result is unsurprising. (18) seems more problematic.[8] I find that the most natural interpretation for (18), even with heavy focal stress, is one where the group of men and the group of women are separate, and that it is only the women who are visiting from Boston. Attachment sites are not recreated in the current framework, but I return to the question of structure and ellipsis below.

Rochemont and Culicover (1990) note that in cases where there is extraposition both from the object and the subject, the clause extraposed from the object must come first.

(19) a man who had blonde hair came into the room last night that I had just been painting.
(20) a man came into the room last night that I had just finished painting who had blonde hair.
(21) *a man came into the room last night who had blonde hair that I had just finished painting.

They attribute this to some form of interpretive nesting requirement, as developed in Fodor (1978) and Pesetsky (1982). Such a requirement can straightforwardly be derived in a dynamic model of syntactic structure building by general principles, invoking the restricted way the pointer can move in the course of a derivation.

4.2.2. Restrictions on the Noun Phrase

The biggest set of restrictions on the acceptability of extraposed relative clauses involves what might be considered semantic properties. However, these do not seem to be based in either structural or lexical properties but can only be fully determined in context. These restrictions concern both the nature of the noun phrase itself with which the relative clause is associated, and the 'meaning' of the predication involved.

[8] Either the initial 'although' clause or the final relative clause seems to require parenthetical intonation. In the latter case, the relative is non-restrictive, picking up both 'the men' and 'the women' as antecedents but not requiring any relation between the two. See chapter 5.

4.2.2.1. (In)Definiteness and the Noun Phrase

Ziv and Cole (1974) note that there is a difference in acceptability between examples such as (22), where the antecedent is indefinite, and (23), where it is definite.[9]

(22) a. a lecturer who was carrying a briefcase arrived.
b. a lecturer arrived who was carrying a briefcase.
(23) a. the lecturer who was carrying a briefcase arrived.
b. the lecturer arrived who was carrying a briefcase.

They explain this in terms of the function(s) of restrictive relative clauses and what information is provided by the noun phrase. Restrictive relative clauses are said not to be acceptable with proper names because the purpose of a (non-extraposed) relative is to provide information to identify the referent, and a proper name provides sufficient information in and of itself. Definite descriptions[10] similarly provide sufficient information to pick out a unique referent. From this they derive the generalisation given in (24).

(24) Members of the semantic class which includes both proper names and definite descriptions may not be modified by <u>identifying</u> restrictive relative clauses.

The term 'identifying' is important because they draw a distinction between the function of *adjacent* relative clauses, which is to identify, and *extraposed* ones, which 'assertate'.[11] By drawing this distinction in function, however, they create for themselves the puzzle of why *extraposed* relatives should be unacceptable with proper names and definite descriptions, given that they claim the function of extraposed relatives is precisely *not* identificatory. Their explanation is to

[9] I do not mark this in terms of '*' or '?'. While the example might be considered more natural, there is nothing actually substandard about the definite, and in the correct context, this would be entirely 'grammatical'. This is discussed further below. Ziv and Cole do mark such examples as '??'.

[10] It is important to note that for Ziv and Cole, the term 'definite description' includes the configuration definite determiner + noun phrase + restrictive relative, all taken as a single unit.

[11] That is, they provide new information. For details of this see Ziv and Cole (1974). This is related to Firbas' notion of Communicative Dynamism (1957); see also Firbas (1992).

suggest that semantic constraints become generalised by analogy, so that restrictive relatives as a whole become subject to the rule (24); and they therefore adopt a reformulation of (24) such that restrictive relative clauses are not qualified by 'identifying'.

Their general approach seems correct in terms of identifying a correlation between the function and acceptability of extraposed relative clauses and the degree to which the antecedent noun uniquely picks out a referent. However, in contrast to their conclusions, I would suggest that extraposed relatives *can* be used for identification. They do seem to be acceptable with definite noun phrases in cases where they can be interpreted as a continuation of an incomplete definite description.[12] This is shown in examples (25) and (26). Note, however, that in cases where the noun phrase can be assumed to identify a unique referent, such as proper names as in (27), or pronouns as in (28), extraposition is impossible. In these cases any form of restrictive relative is impossible, as (29) and (30) demonstrate.

(25) the woman showed up who kissed Laura yesterday.
(26) the man arrived who we had been talking about.
(27) *Patricia arrived first who is driving the Alfa.
(28) *she who is driving the Alfa arrived first.
(29) *Patricia who is driving the Alfa arrived first.[13]
(30) *she arrived first who is driving the Alfa.

---

[12] I see no reason why assertation and identification should be mutually exclusive. The main verb may be low in Communicative Dynamism, hence apparently not assertative, as its function is as an introductory device, whereas the relative may be used in conjunction with the noun phrase to give a full description of the thing being introduced. This is taken by some to be a general characteristic of extraposition.

[13] This is of course acceptable as a non-restrictive relative.

(i) Patricia, who is driving the Alfa, arrived first.

I eschew the standard approach of using the 'that' relative (ii), (iii) to enforce a restrictive construal, as these do not constitute the standard relative construction in my dialect. See Ziv and Cole (1974) for arguments as to why non-restrictive relatives cannot be extraposed.

(ii) the woman that is driving the Alfa
(iii) *Patricia that is driving the Alfa

4.2.2.2. Names and Quantifier Phrases

Guéron (1980)[14], takes up the issue of how the degree of definiteness of the noun phrase interacts with the acceptability of extraposition. She rejects the possibility that this can be determined by any +/- definite feature on the determiner of the noun phrase, casting her explanation instead in terms of thematicity. Specific indefinites and referential definites are described as thematic.[15] May (1977)[16] treats these as Names, to be distinguished from quantifying phrases.[17] Guéron gives the following rule to distinguish the two categories (1980:667).

(31) NP Interpretation
   a. A Name is a complete referring expression. It designates a unique object or individual (or set of these) in the world of discourse, either directly, through the use of proper names or deictic expressions (John, that man), or indirectly, by means of complements containing direct referring expressions (the girl who sits next to you, some of those books).
   b. A Quantifying Phrase is an operator ranging over a set of entities. It does not designate a unique individual.

Extraposition is only available when the noun phrase is a quantifying phrase.[18] While it is not surprising that Names and Quantifier Phrases should behave differently in various syntactic contexts, what is more problematic, and yet demonstrated by solid empirical evidence, is the apparent unpredictability of noun phrases with regard to this property. This is not correlatable with a feature marked on the determiner: Guéron proposes that indefinite noun phrases may be specific, and hence Names, whereas definite noun phrases may be attributive rather than referential, and hence Quantifier Phrases. She gives the following

---

[14] Guéron (1980) deals with prepositional phrase extraposition. The arguments presented there transfer readily to wh- relative extraposition and have generally been assumed to apply to these. Guéron and May (1983) extends the approach to relatives. As I have already noted, extraposition of PPs and of relatives is generally treated as the same phenomenon.

[15] For further discussion of the syntactic impact of these see Enç (1991).

[16] This work is based on the treatment of noun phrases in Partee (1972) and Donnellan (1966).

[17] As Guéron notes, this is claimed to have a major impact on LF, where quantifier phrases raise. See May (1985). For a critique of this whole approach, see Lappin (1991 and 1994).

[18] See also Guéron and May (1983).

examples to illustrate how this can only be determined in context (her 103 and 104).[19]

(32) a. Mary wants to marry a Swede. (ambiguous)
b. She met him in Paris. (specific)
c. ...although she's never met any. (non-specific).
(33) a. I would like to read the review of John's book. (ambiguous)
b. ...that's lying on the table over there. (referential).
c. ...if you ever write it. (non-referential).

It becomes something of a moot point whether this should be characterised as a syntactic or semantic phenomenon. While this characterisation may relate to the eventual interpretation which is derived, these are not inherent properties, and therefore cannot be used as the basis for a grammatical explanation.

4.2.2.3. Interpretation and Structure

While Guéron's analysis seems to imply that extraposed relatives can only modify internal to the noun phrase,[20] Rochemont and Culicover (1990) pursue the opposite tack. An extraposed relative clause is a modifier of its antecedent. However, they claim that in contrast to regular relatives, the extraposed relative is not a constituent of the noun phrase. Rather than being detached through movement, the extraposed relative is base generated in its S-structure position. It is related to the maximal projection of the noun phrase, rather than to some subconstituent of it (the maximal projection of the NP either c-commands or is c-commanded by the extraposed constituent). The syntactic structure they propose is such that the extraposed relative has to modify the entire NP. An extraposed relative cannot be used as an additional modification of some N' level. The general constraint by which extraposed relatives are deemed acceptable or not remains semantic:[21] if a definite picks out a unique set, it is

[19] Cf also the discussion in Kartunnen (1969).
[20] Here, as throughout this thesis, I use the term 'noun phrase' to denote the entire noun phrase complex, ie what is often called the determiner phrase.
[21] The semantic explanation they suggest (following Montague (1974)) is that a relative clause denotes a property, as does a common noun, so a noun phrase modified by relative gives a conjunction of properties. A proper name does not denote a property and hence cannot combine with a relative. While I believe this sort of approach is basically correct, the formulation they

similar to a proper name and this description cannot be modified further. This property is reflected in the syntactic structure. When a definite determiner is used pragmatically to indicate that a referent is not new to the discourse, then it is acceptable with extraposition. Rochemont and Culicover give (34) as an example of this (1990: 60).

(34) that man came into the room that I was telling you about.

In this case the extraposed relative is still not modifying the internal structure of the NP: when 'that man' is understood contrastively, and when therefore, the extraposed relative would be modifying some N' description, they claim that extraposition is unacceptable.

As further evidence they cite (35) and (36), their (4) and (5). Their argument is that the examples in (35) are ambiguous as to whether the adjective takes scope exclusively over the noun or over the noun and the relative.

(35) a. a former marine who was assigned to Paris during the last war just came in.
b. an alleged physician who gave illegal prescriptions to his patients was at the party.
c. a big mouse that was raised on beer is in this box.

Thus (a) is said to be ambiguous between whether the marine was formerly assigned to Paris or whether the assignment to Paris is independent of being a marine. In (b) the question is whether what is alleged is that the person is a physician or that the person is a physician who gave illegal prescriptions. In (c), the two possible readings are that the mouse is big, and that it is big for a mouse that has been raised on beer. The claim is that in (36), with the extraposed structure, the reading where the adjective takes scope over the relative is ruled out.

---

give is too generalised: for example it would rule out non-restrictive relatives on semantic grounds. I discuss this below.

(36) a. a former marine just came in who was assigned to Paris during the last war.
b. an alleged physician was at the party who gave illegal prescriptions to his patients.
c. a big mouse is in this box that was raised on beer.

These data seem to me marginal; the readings for (35) where the adjective takes scope over the relative are somewhat forced. With a considerable effort, such readings could also be imagined for the extraposed examples in (36). I consider that the problem here is the general one of construing an adjective with the relative. What these illustrate, however, is the extent to which Rochemont and Culicover differentiate extraposed and regular relatives. These are modelled not as equivalent, and may therefore differ in interpretation. This departs from approaches where identical interpretation, at least in truth conditional terms, is taken to be definitional of extraposition. The distinction that they are drawing here seems to me to be the one between restrictive and non-restrictive relatives.

This distinction is an important one to make inasmuch as what might be called the non-restrictive case is typical of extraposed relatives; however such a characterisation does not exhaustively describe extraposed relatives. On the one hand, the solution that they propose is neat in that it allows a ready structural explanation of why a definite description cannot be used with an extraposed relative - the relative should be a modifier, but it can only modify the maximal projection of N as it is base generated in a c-commanding position. This means that this can only be interpreted when it is possible to give some *additional* specification of the noun phrase - but that would seem to be a discourse phenomenon. Moreover, it seems that the syntax will not reflect the semantic distinctions.

On the other hand, if extraposition is generally to be treated as base generated in the manner proposed, there is no way to derive quantified phrases. These cannot be reduced to a relation with the maximal projection of the noun phrase. Contrary to the claim that the extraposed relative can never be within the scope of the determiner, there are clear cases where the relative certainly can be within the determiner's scope. In examples with quantifiers, the interpretation is clearly such that the relative clause is bound by the force of the quantifier.

(36) every boy came to the party who I had invited.
(37) most students left who had failed the exam.

In the cases above it is clear that the quantifier binds the variable both in the common noun and in the relative. The meaning of (36) is not that every boy came to the party and I had invited them, as two separate units. However, that would be the only meaning that a Rochemont and Culicover-type structure could allow. Rather, the interpretation is that all of the boys in the set of those that I had invited came to the party. This could be represented as (38).[22]

(38) $\forall x((boy(x) \ \& \ invited(x)(I)) \rightarrow (came\_to\_the\_party(x)))$

The same holds for example (37). The truth conditions are determined by the fact that 'most' ranges over both students and exam failers. Consider a situation in which there are 100 students. If 80 students left, but all of these students have passed the exam, and the 20 remaining students all failed, (37) will not be true, although it would be true to say that most of the students left. If 40 students left, then it is not the case that most of the students left. If, however, amongst this group of 40 were 32 students who failed the exam, and in the set of 100 students overall there were 37 who failed the exam, then (37) certainly holds true.

The heart of the problem really is the matter of interpretation, and how this is related to the structure. I suggest that the problem with extraposed relatives is precisely that they can have either interpretation - that is, they may be used as a restriction on (part of) the noun phrase or they may relate to the noun phrase as a whole. This seems to require two different configurations, but if both are necessary there is no way of choosing one over the other as a putative correct analysis. In transformational terms, this seems to result in a situation where extraposition may be either base generated or created by movement, which is clearly not a satisfactory result. In 4.3. I demonstrate how a dynamic approach allows both structures to be derived by the <LINK> operation while maintaining the interpretive difference.

[22] I explore in greater detail below the semantic structures that might be derived for these examples.

4.2.2.4. Underdetermining Semantic Relations

A plausible way to avoid the structural alternatives and theoretical complications necessary to incorporate extraposed relatives in the grammar is to suggest an analysis where the syntax underspecifies the interpretation. Such an approach is considered by Wittenburg (1987) who examines the way that extraposed relatives can be accounted for in (early) Discourse Representation Theory (Kamp ms.).[23] He is concerned with the way in which an extraposed relative may pick up on any one of several possible antecedents. The example he gives is (39), (his (10)).

(39) a playwright gave a play to some producer that had never been in New York before.

In order to maintain a relatively simple syntax, he suggests that compositionality should be relaxed so that the syntax be allowed to underdetermine the eventual semantic relations, with an extraposed phrase assigned structure simply as a sister to S, or to VP. The semantic dependency is then determined as an anaphoric choice. (40) is the construction algorithm that he gives for extraposed relatives.

(40) Construction Rule for Extraposed Phrases (Wittenburg 1987: 434)
Form of α:[$\beta\gamma$], where β is an S or VP and γ is a PP or R

Operations:
1. Process β, augmenting $U_{k'}$ and $Con_{k'}$
2. Add x to $U_{k'}$
3. Add $\gamma(x)$ to $Con_{k'}$
4. Reduce γ, if necessary
5. Add $[x = y]$ to $Con_{k'}$, where y is in the set of reference markers introduced by β, y is structurally accessible from x, and within this set, y best accommodates the descriptive content of γ.

[where k is the DRS for the current point in the discourse,
k' is the updated DRS,
U_k is the set of discourse markers,
Con_k is the conditions applying in k].

[23] For a more recent general introduction to this see Kamp and Reyle (1993).

According to this procedure, the processing of the sentence or the verb phrase has to be completed before the extraposed relative can be processed. In the case of an indefinite description, the process proceeds unproblematically: a new discourse marker is introduced, and the verb phrase is used; then the relative adds a further condition on the discourse marker to the DRS; this is independent of the initial operation. However in the case of definite noun phrases the situation becomes problematic. A definite description identifies a discourse marker with something already present, according to the descriptive content of the NP. However, when there is extraposition of a relative clause not all of the descriptive content is available locally: the contradiction is reached that the noun phrase cannot be fully processed until the content of the extraposed relative has been added, but that the extraposed relative will not be reached until the noun phrase and the verb phrase have been processed.[24]

The differences in acceptability between definite and indefinite noun phrases is due to the semantic processing procedure. Wittenburg gives the following examples of the gradedness of acceptability judgements (his 24-26).

(41) a. a man won that had never played lotto before.
b. the man won that had never played lotto before.
(42) a. a man won a million dollars that had never played lotto before.
b. ?the man won a million dollars that had never played lotto before.
(43) a. a man gave Ralph a million dollars who he had never seen before.
b. ??the man gave Ralph a million dollars who he had never seen before.

The more intervening material there is, the more difficult it is to process the definite noun phrase. To me this suggests that not only semantic processing has to be a factor, but that more general processing considerations play a part.

The greatest point of dispute concerning extraposed relatives involves how they relate to their head - whether they should be some modification on an already established entity or whether they provide some additional description to identify that entity. Much of the debate has centred on extraposed relatives

[24] This of course leaves a problem: Wittenburg suggests modifications to the DRS construction procedure. I am unaware of any accounts which develop this. As outlined below, this can be solved from the perspective of uniting syntactic and semantic processing.

with indefinite descriptions, taken as the paradigm case of extraposition; but these are the least illuminating with regard to this question. The data is sufficiently debatable to suggest that having both (radically different) operations is an undesirable option. However the disparity of derivations necessary is a function of other assumptions about grammar. In 4.3. I demonstrate how the <LINK> operation can account for both cases and how the correct interpretation is predictable from (but not rigidly determined by) the properties of the antecedent noun. First I examine further evidence concerning the context dependence of 'semantic' restrictions.

4.2.3. Restrictions on the Predicate

In this section I discuss another 'semantic' restriction on extraposed relatives which proves not to be absolute - the type of predicate they can be used with. This is dependent on the context of use. As may be noted, the range of examples of extraposition presented thus far has been limited in terms of the type of predicate used in the main clause. The examples given are typically 'arrive', 'come', 'appear'. I have restricted the examples to these as they are straightforwardly acceptable with extraposed relatives. Now I consider this aspect of extraposed relatives - what restrictions there are on the predicates they can be used with. This has been identified as a constraint whereby extraposed relatives can only appear with 'emergence' predicates. There are two reasons to reject so rigid a characterisation - firstly, as has been generally noted, this is not a semantic constraint in the sense of being a property of the lexical item, but rather is a property of the lexical item in a particular context of usage; secondly, this is characteristic only of a subset of cases of extraposition.

4.3.2.1. Emergence

(44) gives examples of extraposed relative clauses with 'emergence' predicates.[25] Other verbs are not so readily acceptable with this construction. Compare the examples in (44) with those in (45), which indicate transitive actions.

[25] The distinctive syntactic properties of such predicates is discussed in Firbas (1957).

(44) a. a man appeared who was wearing a track suit.
b. a man arrived who I know from the gym.
c. a man turned up who can speak Creole.
(45) a. a man ate the beetroot who was wearing a tracksuit.
b. a man wrote the article who I know from the gym.
c. a man piloted the aeroplane who can speak Creole.

Although I would hesitate to judge the examples in (45) to be ungrammatical, or even marked, they seem to differ from those in (44).

Guéron (1980) notes, in the context of prepositional phrases, that extraposition is only acceptable when the verb can be construed as *merely* asserting the 'appearance' of the subject, but that this cannot be analysed as an intrinsic lexical property of the verb.[26] Thus, she says that (46a) is acceptable, but not so (46b).

(46) a. a little girl skipped past with long braids.
b. *a little girl skipped right by with long braids.

4.2.3.2. Construability

This point is taken up by Rochemont and Culicover (1990: 65-68). They note that while 'scream' seems odd as the matrix predicate in (47) (their 14(b)), it is considerably improved by putting it in the context in (48) (their 15).

(47) a man screamed who wasn't wearing any clothes.
(48) suddenly there was the sound of lions growling. Several women screamed. Then a man screamed who was standing at the very edge of the crowd.

They conclude that the predicate must be c-construable, where this is defined as context-construable, old/given information, material under discussion in the context at hand. Anything that is not c-construable is a focus (1990:20).[27] This property marks out extraposition amongst the group of structures which give

[26] See also Rochemont (1978).
[27] See also Culicover and Rochmemont (1983), Rochemont (1986).

presentational focus. What emerges from their discussion is that the 'emergence' restriction has to be accounted for functionally.

It may be the case that the more c-construable the main predicate is, the more acceptable the extraposed construction is. However, it is not clear that this even has to be a property of the predicate itself. Consider example (49).

(49) the man wrote the article who I know from the gym.

Intuitively this would seem better in a situation where some previous mention has been made of 'the article'; in general cognitive terms this would make the concept of 'writing' more accessible, but this is certainly a matter of pragmatics and not of grammar.

4.2.3.3. Relating the Clauses

It is not just the case that acceptability may depend on the predicate in the main clause. What also seems to be pertinent is the predicate in the extraposed relative, and the relation between the two predicates. There is a contrast between (50), which seems perfectly natural, and (51), which in the absence of any particular context seems to lack a relation between the two clauses. If more biasing material is put in, the contrast becomes sharper. (52) is fine, but (53) seems distinctly odd.

(50) a man translated the article who could speak Creole.
(51) a man translated the article who was wearing a tracksuit.
(52) a man interpreted between the Finns and Swedes who spoke both languages.
(53) a man interpreted between the Finns and Swedes who spoke Chinese.

Ziv notes this contrast specifically in the case of relatives with generics as their antecedent. She talks of this type of extraposed relative as being a '...non-identifying proposition bearing some logical relation to the proposition in the main clause when extraposed' (1975:568). [28] Why the two clauses juxtaposed in

[28] Ziv and Cole (1974) discuss a similar distinction between two types of non-restrictive relatives - appositives, as conventionally understood, and continuatives, which 'usually occur at

this way have to have some type of connection is certainly a pragmatic problem, but the fact that this should be the case in an apparently unitary structure is interesting in itself. In the present approach, where two propositional structures are derived, this is to be expected. In transformational terms, where a premium is placed on the autonomy of the syntactic component, and in terms of a model of utterance interpretation relying exclusively on post-linguistic operations, this is extremely surprising.

What is also notable about the above examples is that they do not seem to involve presentational focus. If the function of extraposition were presentational focus, either of the NP or of the NP plus the extraposed relative, then it would be unsurprising that there should be a 'coming into play' restriction. This might also tie in with the apparent greater acceptability of indefinite noun phrases versus definite noun phrases with extraposition. However, in the above examples, (50)-(53), extraposition with an indefinite is not associated with presentational focus, nor with an 'emergence' predicate. Similarly the examples in (45) above *are* acceptable; any deviance is certainly not a matter of grammaticality. What complicates consideration of these structures is that there are tendencies towards generalised behaviour, but these are not absolute. As such, the correlation of form and function seems to be opaque. It was noted above that there is nothing absolute about the function of definite versus indefinite, as the former can be non-referential and the latter specific, so the present situation is unsurprising.

The general pattern is that indefinites are more associated with presentational occurences of extraposition, as in (54), and definites are more associated with either continuative or identificatory occurrences, as in (55). The 'emergence'

---

the end of the sentence, bear some logical (often causal or temporal) relation to the matrix sentence and thus serve as appropriate continuation of that which was said in the sentence to which they are appended.....continuatives are not derived by relative extraposition.' (1974: 776,777). (i) is an appositive, (ii) a continuative.

(i) Terry, who is a physics teacher, arrived late at the restaurant.
(ii) Terry arrived late at the restaurant, who is normally punctual.

This distinction goes back to Jespersen (1927).

predicate restriction is associated with a particular construction, the presentational type, and as would be expected, this is better with an indefinite.[29]

(54) a man arrived who we had been talking about.
(55) the man wrote the book who we had been talking about.

(56) illustrates that when a non-emergence predicate is used, the indefinite is still acceptable, as is the definite with a presentational, shown in (57). This latter result is to be expected in that the 'presentational' predicate itself contains lexical material.

(56) a man wrote the book who we had been talking about.
(57) the man arrived who we had been talking about

It seems, though, that their functions may correspondingly differ.

4.2.4. Differentiating Extraposed Relatives

Some light can be shed on the function of extraposed relatives by the tag question test. I adapt this from Ziv (1975) who uses it to identify the assertative part of an utterance.[30] In (58) the relative clause forms the assertative part; in (59) the main clause forms the assertative part.

(58) a. a man arrived who we had been talking about, hadn't we?
b. ?a man arrived who we had been talking about, didn't he?
(59) a. ?a man wrote the book who we had been talking about, hadn't we?
b. a man wrote the book who we had been talking about, didn't he?

However, these results depend on the stress pattern. The constituent which is stressed is the assertative part and hence it is possible to stress 'arrived' in 58(b) and have the tag question 'didn't we'.

---

[29] It is well established that definites and indefinites behave differently in this sort of context. This goes back to Milsark (1977). Contrast:

(i) there is a man in the garden.
(ii) there is the man in the garden.

[30] This comes from Hooper and Thompson (1973), who use it to distinguish assertions from non-assertions.

The examples below show the same tests with the definite. When the definite is used with an 'emergence' predicate, either tag question is acceptable, suggesting that either the presentational reading or the identificational/continuative reading can be acceptable.

(60) a. the man arrived who we had been talking about, hadn't we?
b. the man arrived who we had been talking about, didn't he?
(61) a. ?the man wrote the book who we had been talking about, hadn't we?
b. the man wrote the book who we had been talking about, didn't he?

If both results are possible, then there is nothing in the nature of the emergence predicate which forces the presentational reading. This result can be replicated with indefinites.[31]

(62) a man arrived who had served in the army.
(63) a man arrived who had crashed on the way from London

In (62) what seems the more important information is that the man had served in the army, whereas in (63) the fact of arrival may well be of greater significance. This can only be determined in context.

Contrary to what is generally claimed, in certain circumstances, the definite is more acceptable than the indefinite.

(64) the man ate the melon who we had been talking about.
(65) a man ate the melon who we had been talking about.

Here the problem is precisely that the extraposed relative requires a presupposed antecedent which an indefinite cannot supply, but this is due to the meaning of the predicate in the relative.

Should the apparent difference in informational status be reflected in the structure built? It is not clear: these particular results are inconclusive. In (58) for example, either tag question can be used according to the pattern of stress,

[31] This point is made in Guéron (1980). It is more accurate to speak of emergence readings of particular predicates, these being most prominent when there is no other material in the matrix clause.

but there is no sense in which the 'wrong' stress pattern affects the grammaticality, or, for that matter, ease of interpretation. If the function here is influenced by anything, it must also be sensitive to the content of the matrix clause, in terms of the nature of the determiner and the meaning of the predicate. The most appropriate way to address this is by fitting it into a dynamic model of structural development. Moreover, in developing an account of extraposed relatives, what has to be considered is what actually are *linguistic* properties and what might justifiably be considered to be *non-linguistic* properties; the present framework allows consideration of this not only in terms of those properties that are actual properties of the structure (form) as against those that are associated with the structure (function), but also properties of the *process* by which that structure is derived.

Characteristic of the differences outlined above is that they do not affect the truth conditions of the representation to be derived. But there is another distinction to be drawn, that between continuative and identificational. This does have a bearing on truth conditions. Consider the examples below.

(66) a. a man walked in who was wearing brightly coloured pants, wasn't he?
b. ?a man walked in who was wearing brightly coloured pant, didn't he?

(67) a. the man walked in who was wearing brightly coloured pants, wasn't he?
b. the man walked in who was wearing brightly coloured pants, didn't he?

(68) a. every man walked in who was wearing brightly coloured pants, didn't he?
b. *every man walked in who was wearing brightly coloured pants, wasn't he?

In (68), there is a clear difference between the (a) and (b) versions: (68b) is strongly unacceptable. As has already been discussed above, in cases which involve quantification, the quantifier has to take scope over the relative clause. Any syntactic input which does not reflect this will yield the wrong semantic interpretation.

In summary, the problems presented by extraposed relative clauses and deciding when they can be well-formed constituents of an utterance seem to range over a wide and disparate set of theoretical areas, transcending neat borders of classification. In Labelled Deductive Systems for Natural Language such divisions are drawn differently. I propose an account in this framework which derives their idiosyncrasies from the process of structure building and incorporates pragmatic choices through underspecification.

The possible functions of the extraposed construction seem to be:

i. presentational/assertative: the extraposed relative provides the main information in the utterance;
ii. identificatory: the extraposed relative provides a continuation of the description of the noun phrase;
iii. continuative: the extraposed relative provides an additional assertion about a noun phrase already identified.

These are not exclusive: (i) is compatible with (ii) or (iii). (i) is a function of informational status and the interaction with context. The presentational function is achieved by the combination of a (novel) noun phrase with a basic 'emergence' predicate and a stressed relative clause. In this case the reason for extraposition is that, other things being equal, if the relative were not extraposed and the emergence predicate were the final constituent in the utterance, it is this which would appear to be the most important information rather than the relative.

The significant difference is between (ii) and (iii). These may differ in terms of truth conditions and implications; and this has to stem from different structural representations. As suggested at the end of section 4.2.1., this is the difference between restrictive and non-restrictive interpretations, which can be reconsidered in terms of the structures derived by the <LINK> operation.

4.3. Extraposition and Type (t) <LINK>

4.3.1. Distinguishing <LINK> Structures: Restrictive and Non-Restrictive

I will discuss in general terms the restrictive/non-restrictive distinction before detailing how it applies to extraposed relatives.

The distinction between restrictive and non-restrictive <LINK> structures was briefly outlined in chapter two. The point I must first make is that this differs from the standard understanding of this distinction with regard to relative clauses, shown in the examples below.

(69) the man who teaches syntax ate a bagel.
(70) John, who teaches syntax, ate a bagel.

The difference in interpretation is clear: in (69) teaching syntax is used as qualification of the noun phrase, in (70) it is not; that is to say, in (69) the relative restricts the interpretation of the nominal complex, whereas in (70) the interpretation of the noun is already established and the relative clause, rather than restricting it, just supplies another piece of information. Hence the term 'non-restrictive'. The other difference between these two examples is that (70) is a parenthetical.[32] The two properties, being non-restrictive and being parenthetical, have generally been conflated, but there is no reason why this has to be the case. A parenthetical relative clause will always be non-restrictive,[33] but that does not imply that the converse has to be true: there is no reason why a non-parenthetical relative clause has to be restrictive. This is the key to the behaviour of extraposed relatives; at the same time, it is the processing ambiguity that makes them so apparently 'badly behaved', in terms of subjectability to neat classification.

Labelled Deductive Systems for Natural Language seeks to unify syntactic and semantic aspects of linguistic behaviour. The restrictive/non-restrictive distinction has implications for both, in terms of the mapping between the two levels assuming that interpretive differences are reflected in the structure, and with regard to the differences that these structures display in purely formal

[32] These are discussed in detail in chapter six.
[33] Depending on the definition of parentheticals. See chapter six.

terms. The analysis to be given is that a uniform operation can be applied to derive the representation for relative clauses, but the location where it operates determines the semantic content; this process of deriving the structure results in the syntactic properties manifested. The restrictive/non-restrictive distinction is used to distinguish semantic objects - that is, the output representations of the system will differ, and the process of getting to that representation also differs. This distinction can be described as a property of L-trees, but this property derives from their constituency; it is effectively a meta-description, rather than being intrinsic to the nature of a syntactic operation.

The <LINK> operation launches a new processing task, with its own individuated goal, to build a tree of type (t). By definition, this has to share a node description with some node in the tree from which the <LINK> operation was launched. The difference between the restrictive L-tree and the non-restrictive L-tree stems from the type of node which is shared: where that node originates and what content it has.

As discussed in 2.7.2., in a restrictive L-tree, the node which is shared between the L-tree and the main tree is a metavariable from some cn-task in the main tree.

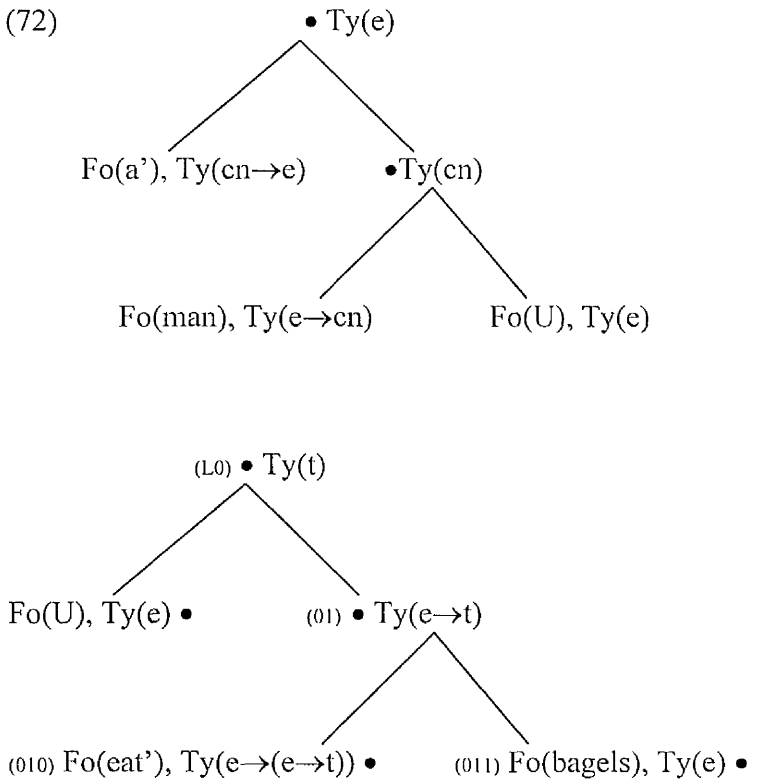
In a non-restrictive L-tree, the node which is shared between the L-tree and the main tree is a completed e-node, that is, a node which itself has internal structure but where that structure is not accessible, and is not affected in any way by the L-tree.

This reflects different steps in the *process* of structure building. In the restrictive case, the original e-task has not yet been completed. There is still the overall requirement TODO type(e); the L-tree is part of the description within that goal. The end result is a composite semantic unit, where any restrictions binding the variable in the common noun will equally bind the variable in the relative clause. This stems from the fact that the linked node is an exact copy: effectively these are two instantiations of one variable. The requirement of type (e) can only be fulfilled after the processing of the <LINK> structure. This is illustrated in (72), the structure built for (71) after all the lexical premises have

been accessed. For the sake of clarity in these tensed examples I suppress representation of the p-node and the tense node.

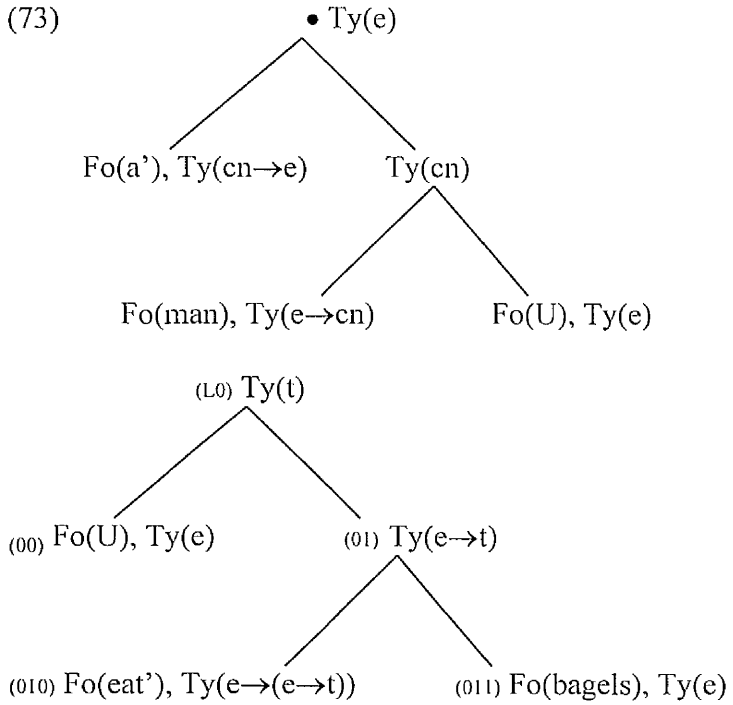
(71) a man who eats bagels

(72)



Processing of this proceeds directly as the lexical information is accessed, but note the indicated task states. Compilation, ie Completion and Elimination, does not occur until all the items in the string have been incorporated into a well-formed tree structure. The local e-task is not satisfied until after the <LINK> tree has been processed. (73) shows how this happens: the premises in the <LINK> tree are combined to give a tree of type (t) (though the restriction on the metavariable is as yet not established). Then the pointer returns to the original tree, where the metavariable combines with the predicate.

(73)



This is duly combined with the determiner to complete the e-task, giving the representation in (74). The article 'a', translated as an epsilon term, restricts both occurrences of the variable.[34, 35]

(74) (ε, x, (man(x) & (eat(bagels)(x))))

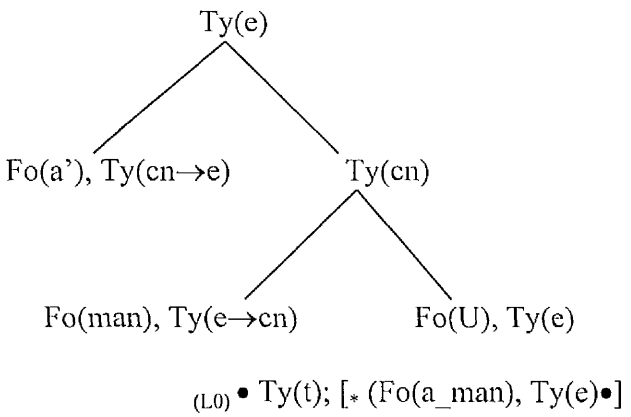
In contrast to this is the non-restrictive case. Here the processing of the e-task is completed before the L-tree is built. Completion and elimination have taken place within the e-task, and the node is duly annotated with the description type (e). It is the whole of the content of the e-node which is carried across into the L-tree, rather than some part internal to it. The processing task instantiated by the <LINK> operation is entirely outside and independent of the e-node. Nothing within the e-node will bind anything in the L-tree and the L-tree will have no effect on the content of the e-node. (75) gives a non-restrictive used with an indefinite noun phrase

[34] Quantification in LDSNL is discussed in more detail in 5.2.3.
[35] Conjunction and type differences are discussed in chapter six.

(75) a man, who eats bagels

As shown in (76), all the processing in the initial e-task is complete *before* the <LINK> operation occurs.

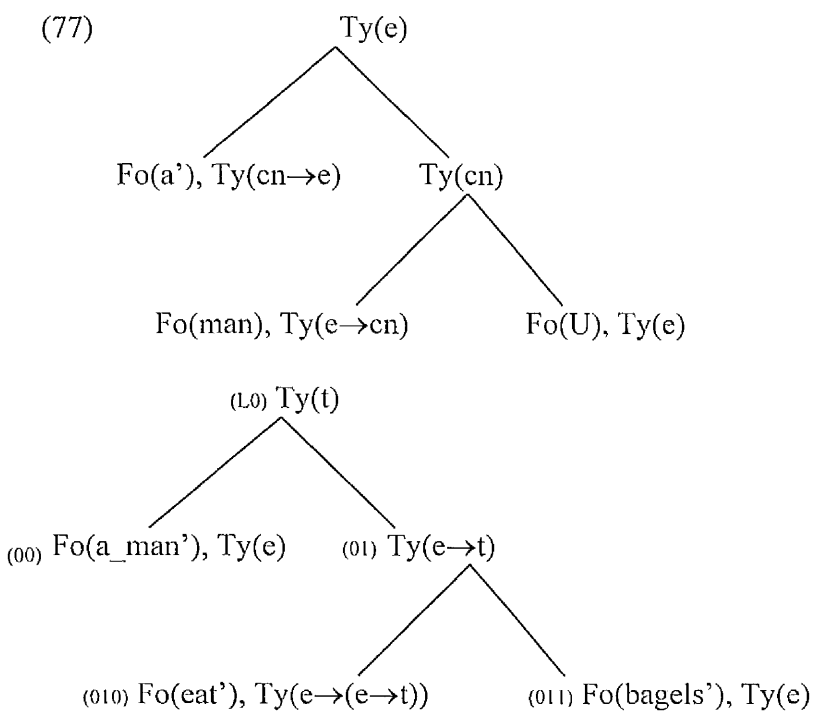
(76)



This is why the non-restrictive reading is associated with so-called comma intonation: the pause indicates that the current processing task (the e-node) is completed (though the *overall* goal of building a tree of type 't' containing that e-node remains).

This results in the overall representation given in (77), interpretable as (78).

(77)



(78) (ε, x, (man(x)) & (eat(bagels)(ε, x, (man(x))))

How the instantiation of the copy of this node should be conceptualised is an interesting question in and of itself, ie what it really means to say 'the whole of the content of the node'. Here I can only speculate. It could be suggested that once completion and elimination have occurred, some form of 'discourse marker' is introduced to the representation, along the lines of those in Discourse Representation Theory (*inter alia* Kamp and Reyle (1993)). However, this remains a linguistic object in terms of tree construction. Although the internal linguistic structure of the entity may be opaque, the e-node itself remains a linguistic representation. Given the sensitivity of subsequent tree development to already established information, it may be that the *conceptual* content is accessible; ie that information represented by the formula description annotated on the node. In other words, once completion and elimination have occurred for an e-task, at that location the representation of a concept is made accessible (according to information supplied by the formula), while the possibilities of linguistic combination are further constrained by the type information associated with that node.

Restrictive and non-restrictive occurrences of the <LINK> operation do give different structures with different truth-conditional interpretations. The different functions attributable to the two sorts, as representations of relative clauses, stem from the dynamics: an e-node identifies an entity, so any operation within that e-task will contribute to the establishment of that entity. If an e-node is complete, the entity has been established and the L-tree provides an additional assertion about that.

The difference between the two as set out depends solely on where in the *process* of structure building the <LINK> operation is launched. That is why it is not necessary that a non-restrictive L-tree be a parenthetical element. All that is required is that the node carried over is the result of a completed e-task. In the case of continuous processing of noun plus relative, the metavariable will be carried across as a matter of course. Parenthetical intonation facilitates the non-

restrictive process: it is necessary to cause completion and elimination to occur, and the e-task to be completed. However, when there is not continuity of input, there is no a priori reason to carry across the metavariable as opposed to the e-node. Hence extraposed relatives should occur both as restrictive and non-restrictive L-trees.

4.3.2. The <LINK> Operation and Discontinuous Input

The definitions involved in <LINK> operations were originally developed for continuous input (ie adjacent constituents) (cf Kempson, Meyer-Viol & Gabbay (in prep.), Kempson and Edwards (1998)). It is therefore necessary to modify these in order to extend the account to cases of discontinuous input.[36] In changing these definitions it should be possible to maintain them as general forms of the rule, rather than having to state separate rules for continuous and discontinuous instances of L-tree building.

4.3.2.1. Redefining the <LINK> Rule

(79) gives the <LINK> rule, as stated in chapter two.

(79) The <LINK> Rule for Tensed Relatives

$$\frac{[_m \ldots\ldots\ldots [_n \, Fo(\alpha),Ty(e)] .. \bullet \, Ty(t)]}{[_m \ldots\ldots\ldots [_n \, Fo(\alpha),Ty(e)] .. \bullet \, Ty(t)], [_{nL} \bullet \, Ty(t); [_* \bullet \, Fo(\alpha),Ty(e) \,] \,]}$$

The problem with this formulation of the <LINK> rule is that it requires that the node from which the <LINK> operation is launched is the node which is carried over into the L-tree. Moreover, it brands the L-tree with the location of that node. The effect of this is to impose a requirement of strict adjacency. Such a requirement precisely does not apply in cases of discontinuous input, where this rule would either give the incorrect structure or would fail to apply at all. If the immediately preceding node is of type (e) then this <LINK> rule can apply, but

[36] I use this term in presenting the LDSNL account to reflect the particular perspective of the framework: the problem here is how to build structure for constituents which though not part of the same immediate processing task nevertheless require dependencies to be modelled between them.

the node carried over into the L-tree will be the wrong one. For example in (80), application of the above rule would derive the structure where the relative is modifying 'my sister', but not where it is modifying 'the man'.

(80) the man talked to my sister who I was telling you about.

However, both readings are available.

If the immediately preceding constituent is not of type (e), the rule cannot even apply, as is the case for (81).

(81) The man arrived who John knows.

The problem with this formulation, then, is twofold - that the tree update rule has to apply only at locations of type (e), and that the content of the node at that location then has to be carried over. The solution is to remove these restrictions, so that it is possible to launch the operation at any point in the process of tree development and to allow any node from the main tree to be the node which is carried over. The revised form of the rule is given in (82).

(82) The <LINK> Rule for Tensed Relatives (revised)

$$\frac{[_m..[_n\ Fo(\alpha),Ty(x))]..[_o\ Ty(y)]..\bullet\ Ty(t)]}{[_m..[_n\ Fo(\alpha),Ty(x))]..[_o\ Ty(y)]..\bullet\ Ty(t)],\ [_L\ \bullet\ Ty(t),\ [_*\ \bullet\ Fo(\alpha),Ty(x)\ ]]}$$

(where x and y are variables over the set of types).

This rule states that at any location, tree node (o), which is annotated with a type description, and where at least one node is annotated with a description, it is possible to launch the <LINK> operation.[37] The effect of that is to build an L-tree annotated with the requirement type (t), which has imposed on it the requirement that there be some node held in common with the main tree. This has no effect on any tree relations established in the main tree.

[37] So the type of the launch node has been annotated and is not a requirement, and there is a node somewhere in the tree where a formula description has been annotated.

With this formulation of the <LINK> rule, the function of the wh- element is still to change this node annotation from a requirement to a description, but where that description is underspecified for its location in the tree. (83) gives the lexical entry; this is changed from that in chapter two by removing the node branding on the L specification.

(83) Lexical Entry for Wh- Relativiser

$$\frac{\text{WH} \quad \text{Tr} \quad [_{L}\,[_{*} \bullet \ \text{Fo}(\alpha), \text{Ty}(e)]]}{\text{Ac} \quad f_{\text{DONE}} \ \text{Fo}(\alpha), \text{Ty}(e)}$$

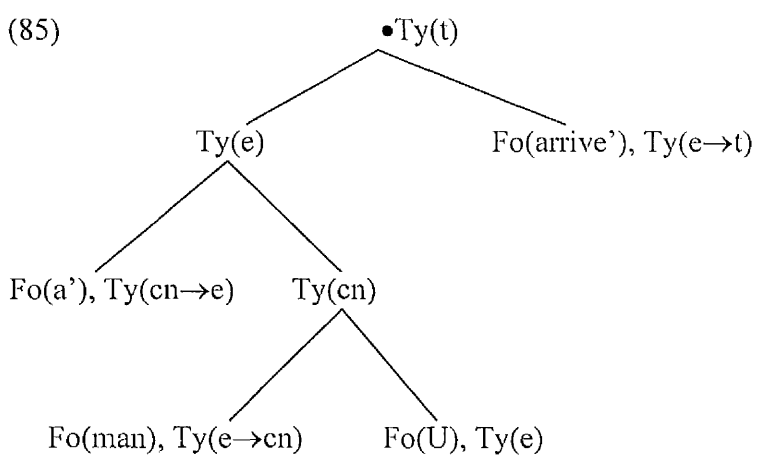
The triggering environment is an underlocated requirement in an L-tree; the action is to change this requirement to a description.

The definitions above guarantee that well-formed tree structures can be constructed for cases of discontinuous input, the 'extraposed' relatives. I will briefly illustrate this, using the example in (84).

(84) a man arrived who I like.

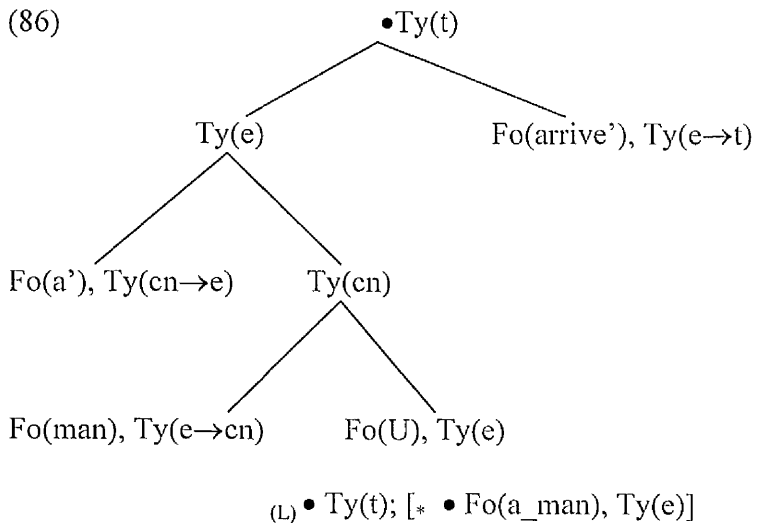
The main clause will be processed standardly to give the structure in (85), assuming that the extraposed relative is to be interpreted as a non-restrictive.

(85)



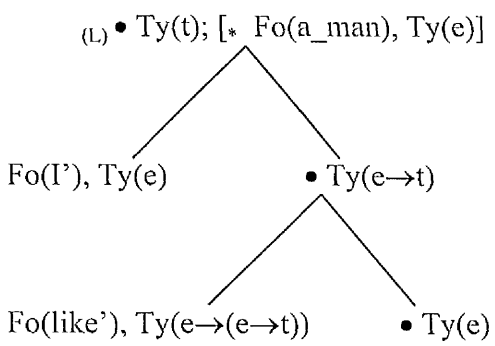
Note that overall compilation has not yet occurred. At this point the <LINK> rule can apply to launch the construction of a new tree, as shown in (86).

(86)



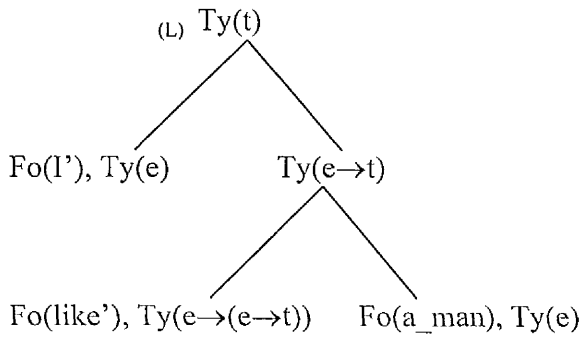
The development of the L-tree proceeds according to the remaining linguistic input. 'Who' changes the annotation from a being a requirement to being a description. Scanning the rest of the input gives the structure in (87).

(87)



The underlocation of the node which is linked is resolved in the object position, where there is a requirement of type (e). Compilation of the tree can proceed to complete the t-task, giving a completed L-tree of type (t).

(88)



The pointer then returns to the main tree. Now compilation can take place there. All of this results in the interpretation in (89).

(89) ((ε, x, man(x) & arrive(x)) & (like(ε, x, man(x))(I)))

To indicate that whatever dependencies have been fixed in the processing of the main clause remain unaltered in the L-tree, I will represent this as (90).

(90) ((ε, x, man(x) & arrive(x)) & (like(a_man)(I)))

What the above account glosses over is the problem of how to choose what node is to be carried over into the L-tree. The link rule itself does not specify what node this should be: it is no longer a matter of this automatically being the immediately preceding node. Effectively the system allows a free choice. There is nothing in principle that would for example stop the (e→t) node from being carried over. What ultimately determine this are the requirements of the L-tree, in that there has to be a match between the node carried over and some requirement in the L-tree. That is to say, there is no point in carrying over a node of type (e→t), if it turns out that what is needed is a node of type (e). This is where the relative pronoun proves crucial.[38]

[38] I do not consider 'that' here for reasons stated. It is possible to define the same sort of lexical entry as for wh- relatives. However, it might be possible to develop a unified semantics for various uses of 'that' as a tree launching procedure. I leave this as a matter for further research.

4.3.2.2. Wh- and <LINK>

The wh- relative effectively acts as an anaphoric device, requiring identification with some linguistic description, and in certain cases providing information to delimit the search for that referent.[39] 'Who' provides the information that the antecedent has to be animate, and of type (e), while 'which' requires that it be inanimate. As presently stated , the <LINK> rule requires identification of the shared node at the inception of the L-tree. In terms of deriving well-formedness requirements this is neither here nor there; however, in terms of developing a model of the incremental processing of input, this seems to ignore any possible role played by the wh- relativiser.[40] The role of the wh- word as a relativiser combines an anaphoric function and the function of supplying an (underspecified) nodal location, where this only happens in an L-tree.

The system allows for context dependent lexical entries, and there is already a distinction drawn between wh- elements as question words and wh- elements as relativisers. Given these facts, it is possible to develop a lexical entry for the wh- relativiser combining all the above facets. (91) gives the definition for 'who' as a relative pronoun. Here it is the lexical entry itself which causes the <LINK> operation to apply, ie the <LINK> operation is encoded within the lexical entry.[41]

(91) Tr $[_m \ldots [_n \, Fo(\alpha), Ty(e)] \ldots [_o \, Ty(y)] .. \bullet \, Ty(t)]$
Ac i[L0],
f_{TODO} (Ty(t)),
i<*>,
f_{TODO} (Ty(e), Fo(WHO)),
f_{DONE} (Ty(e), Fo(U))

The trigger for operation of this lexical rule is the same as was defined for the <LINK> operation in general. It can therefore apply in cases both of continuous

[39] This is essentially the approach proposed in Wittenburg (1987).
[40] Furthermore, this does not allow explicit representation of a notion of a pragmatic choice, which may depend on the content of the L-tree, without having restructuring operations. The metavariable supplied by the wh- word can act as a placeholder.
[41] This has no bearing on the general availability of the <LINK> operation as a processing operation: I am not suggesting that <LINK> has to be encoded in a lexical rule, just that it is on this occasion.

and discontinuous relative clauses. The action in terms of tree update is as follows: build the root node of an L-tree and label this with the requirement of type (t). Build a node which is underspecified for its location within the tree. Annotate this node with the requirement of type (e) and formula (WHO). The formula information WHO here represents two things: a place-holding metavariable, and a search procedure for some entity with the characteristic 'human' to identify with that variable. When that search procedure has occurred, the result is slotted into this node location. The node is labelled with the description type (e) and formula (U), where (U) is a metavariable to represent the result of the search procedure (ie identified with α in the main tree). This node remains underlocated, but that is a matter of resolution as the tree develops, not specific to anything about the present construction.

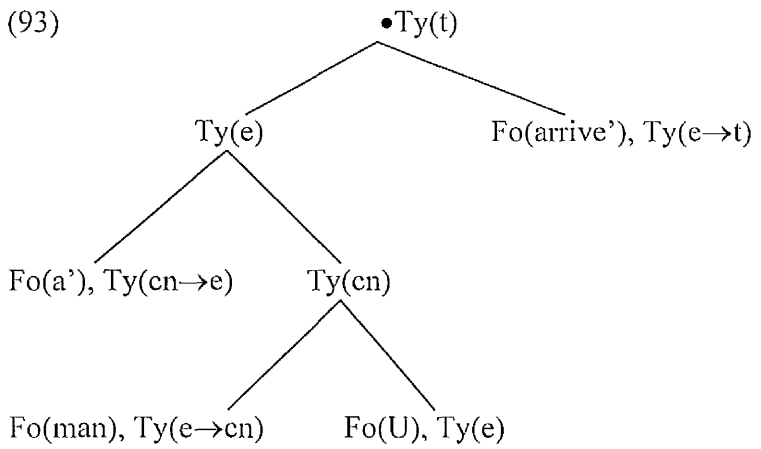
4.3.2.3. Non-Restrictive L-trees

The sample derivation was given for (84), repeated below as (92).

(92) a man arrived who I like.

The definition for 'who' above alters the construction process in that: i) the <LINK> operation is induced by lexical specification and ii) the wh- word creates the node which has to share a description with some node in the main tree and instantiates a search for the appropriate content. (93) gives the tree structure as 'who' launches an anaphoric search. (94) shows the L-tree after this has been completed.

(93)



$_{(L)}$ • Ty(t); [$_*$ • Fo(WHO), Ty(e)]

(94) $_{(L)}$ • Ty(t); [$_*$ Fo(a_man), Ty(e)]

This redistribution of the burden of processing tasks results in a more constrained account of the way in which L-trees are derived for relative clauses, while allowing that it may be the case that the relative pronoun allows a free choice of antecedent. The choice of antecedent is subject to evaluation according to pragmatic criteria, specifically those of Relevance.

The above discussion indicates how discontinuous relatives are processed, and how this is facilitated by modelling the wh- relativiser as an anaphoric device. This straightforwardly applies also to cases of continuous input. However, the above derivation involved a non-restrictive, and I have already stated that discontinuous relatives may map onto restrictive L-trees. I now turn to the particular problems that these raise.

4.3.2.4. Discontinuous Input and Restrictive L-trees

In the case of continuous relatives the building of a restrictive <LINK> structure is straightforward: the metavariable from the common noun is picked up on by the relative pronoun and it is this which provides the nodes shared by the trees. This all happens within the e-task; and specifically compilation does not occur until the L-tree has been successfully completed. In the case of a discontinuous restrictive it is not obvious how this can happen.

Superficially it would appear that it is simply a matter of the relative pronoun instantiating its search procedure and identifying the metavariable as the requisite antecedent. This, however, seems problematic: in terms of the tree, the search procedure would first come across the completed e-node, and have this as a possible candidate as antecedent for the discontinuous relative. The non-restrictive interpretation would have to be constructed and rejected before the restrictive one could even be considered.

The problem is more serious. Consider what it means to say that it is the metavariable that is carried over. What characterises the restrictive L-tree is that it provides information which is incorporated within the e-task. In the continuous case this is manifested in the actual order of processing. In the discontinuous case, the e-node has already been completed, yet further information now has to be incorporated into it. This is essentially the problem that Wittenburg (1987) describes for the semantic processing of a definite noun phrase with an extraposed relative - that the noun phrase has to be processed before you can get to the relative but that the relative is necessary to process the noun phrase. In present terms, to specify a further predication on the metavariable is to disrupt established semantic relations and the tree configurations which represent them. Moreover, the system is very precise about task states - there is explicit labelling of the task state at a location. If all the requirements at a node are met then that node is done. It is inconsistent to change the task state - either a node has outstanding requirements or it does not.

The solution lies in modifying the idea of compilation, and how this interacts with tree construction. This in turn sheds greater light on the correlation of form and function.

4.3.2.5. Pointer Movement and Completion

The process of tree construction is driven by information embodied either in specific rules or in lexical entries. These determine the development of the nodes, and thus in turn constrain the movement of the pointer. The pointer indicates where the process of construction has got to. As structure is built, so the pointer moves on. The pointer may also be moved without affecting the

structure, but only as specifically licensed in certain contexts. That is to say that the pointer cannot just be randomly moved across a tree.

As has been outlined thus far there are three ways to effect movement of the pointer:

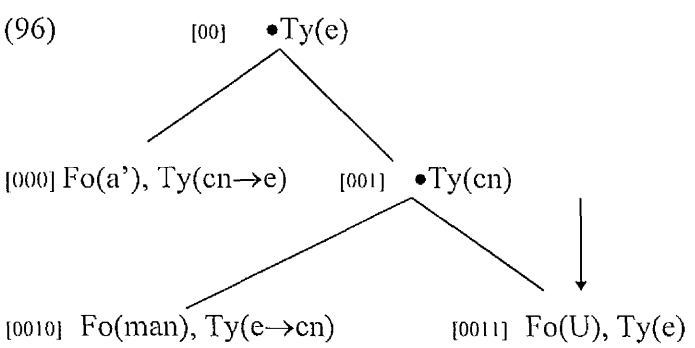
i) through the application of structural rules such as the subject rule in 3.6.1.
ii) through procedures encoded in lexical entries as illustrated in 2.3.
iii) through the application of the rules of Prediction, which allows construction of a node at a child of the current location, and Completion, which allows the description at the current node to be annotated on the parent node.

Completion and Elimination together allow for the process of compilation to take place, to combine the premises to form a semantic object of the required type. Completion allows that a description on a child can be annotated on the parent; Elimination allows the actual combination of premises when the child nodes are annotated with the correct types for type combination (cf 2.2.2.). These rules can apply when all the requirements have been met.

Consider the process of constructing a node with the description type (e), shown for example (95) in (96).

(95) a man

(96)



The overall requirement is for type (e); the determiner imposes the requirement for type (cn). The other requirements are imposed and satisfied locally according to specifications in the lexical entries. It is the combinatorial

requirements which remain to do until Completion and Elimination apply. In (96), all the lexical information has been scanned; the pointer is at (0011). The regular process of compilation is that the common noun and metavariable combine and the pointer then moves up to (001), fulfilling the need for a common noun, and thus changing this from a requirement to a description. This then combines with the determiner to give an entity, and the pointer duly moves back to (00). The task is complete; an entity has been derived which in semantic terms is an evaluable object and which in cognitive terms is an identified representation. The movement of the pointer back up the tree is effected by combining the premises.

What is necessary in the case of restrictives is that combination has not yet taken place. So what is needed is a way of moving the pointer back up the tree without forcing the premises to combine. This can be done using only the rule of completion. I repeat this rule in (97).

(97) $$\frac{[_m \ldots [_n Y\ [_j\ \phi] \ldots]}{[_m \ldots [_n \ldots Y, \langle j \rangle\ \phi, [_j \phi]] \ldots]}$$

where $j = \{u, d, L, *\}$

(97) states that an annotation on the child node can be annotated on the parent node. This makes it possible to maintain a record of all the information on the nodes in a tree, and move the pointer back up without actually combining the nodes. The definition does not state that these annotations have to be descriptions. They may be requirements. Any outstanding requirements will remain, but there is an explicit record of what *may* happen in terms of combining the nodes. So for example, going back to the e-task shown in (96), while there will remain at 001 the requirement of type (cn), there will be explicit labelling to indicate that this requirement can be met by combining the child nodes as and when Elimination does occur. When application of Completion has moved the pointer back to 00, the building of the rest of the tree can continue.[42]

[42] ie the overall tree with requirement type (t), (not shown in (96).

The tree relations internal to the e-task have been fixed (note that these cannot and will not be altered), but the nodes have not been combined. The entity has not yet been established. As the location is still marked with a requirement, further information can be incorporated within this e-task.

This has implications for building structure for discontinuous relatives as follows. The general process of structure building can proceed without requiring that compilation of the internal structure of the node annotated with the requirement type (e) should take place. The search procedure associated with the wh- relativiser can identify the common noun metavariable as a suitable antecedent without interference from the e-node, as no e-node has yet been established. The L-tree can be incorporated into the processing task described by the requirement of type (e), as this remains unfinished. Any restriction imposed by a determiner on the metavariable in the main tree will similarly restrict the metavariable as it appears in the L-tree. This allows restrictive L-trees to be built for discontinuous input.

4.2.2.6. Building Restrictive L-trees

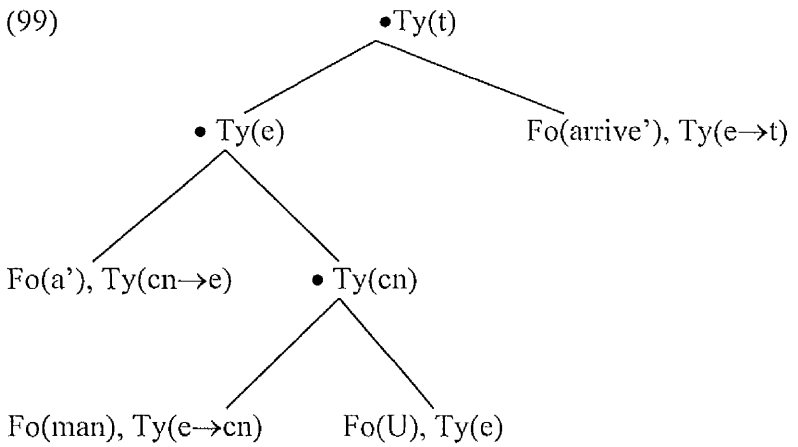
I illustrate this using the example in (98).

(98) a man arrived who I like.

(99) gives the structure built for (98) before the discontinuous relative has been processed. Note that compilation has not yet taken place for the entity and so this remains an uncompleted processing task.[43]

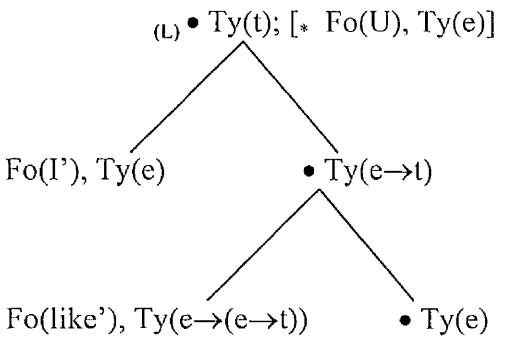
[43] Completion has taken place, but Elimination has not. Given that completion has taken place, all the child annotations are annotated on the parent node; however, as elsewhere, I omit these for reasons of legibility; moreover, this can be derived from the tree representation.

(99)



Scanning 'who' results in the building of a link tree: the metavariable from the common noun being identified as the node to be shared, this remains underlocated until that is resolved by the subsequent tree development. This is shown below.

(100)



When all the lexical premises have been scanned, the result is a well-formed structure, and compilation can take place. The nodes of L-tree are combined as standard; this then returns the pointer to the original location of the metavariable in the main tree. The nodes here are compiled and the overall requirement of type (e) is satisfied. The entity can then compile with the predicate to give a tree with the description type (t) shown in (101).

(101) (arrive(ε, x, man(x) & like(x)(I)))

I now explore the ways this approach can illuminate the properties of extraposed relatives.

4.4. <LINK> and the Properties of Extraposed Relatives

The different properties exhibited by extraposed relative clauses can be explained by the sort of <LINK> structure that they give rise to. Identificatory relatives are restrictive L-trees, which serve to identify the referent. Continuative relatives are non-restrictive L-trees, which provide an additional predication about the entity. As was noted above, there is no unique mapping between certain sorts of determiner and certain sorts of L-tree. Rather, the type of L-tree which is derived for a particular utterance in a context of use will depend on how the noun phrase is processed initially. The distinction is stated in terms of the task state at the e-node. Either the e-task is done or it remains to do, according to whether or not compilation has taken place. This can be correlated to pragmatic factors about whether or not the noun phrase has succeeded in performing a particular function.

As observed in the literature discussed above, there is nothing intrinsic about an indefinite or a definite noun phrase which will determine the properties. Rather this is a matter of what happens dynamically, and the particular context of use. This approach does not require ambiguity in the lexical items; it reflects the way in which the final structural derivation is underdetermined by the lexical information. Structure building is a dynamic process where pragmatic choices have to be built in to the system, which will have effects on the possibilities available later.

I now discuss how the options are delineated according to the type of determiner. In the cases of quantification, the e-task remains to do. For the definite either option is available. In the case of the indefinite, there seems to be no conclusive evidence.[44, 45]

---

[44] I am concerned here only with the behaviour of the indefinite in this specific environment.
[45] I discuss quantification further in 5.2.3. See also Kempson, Meyer-Viol & Gabbay (in prep.).

4.4.1. Determiner and the Dynamics of Structure Building

4.4.1.1. The Definite

In the case of a definite description, either restrictive or non-restrictive L-trees can be constructed depending on what happens at the e-node. Definite descriptions tend to refer to entities already introduced into the discourse. The distinction is clear in that the initial description (ie the predicate supplied by the common noun) either does or does not supply sufficient information to pick out a particular referent.

What characterises the restrictive interpretation is that the task of processing the entity is not yet complete. This is characterisable as an on-line phenomenon. If the definite description does not provide sufficient information to identify a referent, compilation will not proceed and the metavariable will remain free, available for further predication (ie for further restrictions to be made on it). When the wh- relativiser launches the search procedure for the node to be linked, it is the metavariable which will be identified. Whether or not a description is sufficient to identify a referent is dependent on the cognitive resources available to the hearer in a particular instance.

If the input *is* sufficient to identify some referent, then compilation takes place and the task state at the e-node becomes DONE. In this case no restructuring of the internal material is possible, and nothing within that processing task is available for use subsequently in tree building operations. In this situation only the non-restrictive L-tree can be built.

4.4.1.2. The Quantified Phrase

With quantifier phrases the natural interpretation is that quantifier scope does extend to the relative. The L-tree is restrictive; the shared node is the metavariable from the common noun. In this case, compilation does not occur in the e-task and it remains to do. This reflects the fact that quantifier dependencies cannot be established until the rest of the tree has been developed.

4.4.1.3. The Indefinite

Given the above, it seems that indefinites also should not be subject to compilation and that they should be associated with restrictive L-trees. It is not clear, however, that this *has* to be the case. In the case of a specific indefinite which picks out an individual referent, it would theoretically be possible to compile, have the e-task as done and then have this as the shared node in a non-restrictive L-tree. This depends on the analysis of indefinites and existential quantification. Such investigation is well beyond the scope of the present work. Moreover, this does not have any effect on the truth conditions, so it is difficult to apply any diagnostic to the present situation.

On the one hand, indefinites with extraposed relatives are readily associated with the presentational function. Intuitively it would seem preferable that these should be analysed thus: the whole e-node is taken over as this is completed as a processing task, and then the second proposition (the relative clause) carries the main assertative weight of the utterance. When the linked node is the whole e-task then the <LINK> tree should be able to function in the primary communicative function. However, indefinite descriptions tend to be used to introduce novel entities; it is less clear, therefore, whether the material in the e-node should be compiled, as it will not be apparent at what point sufficient information has been obtained to identify the entity appropriately (hence the association of indefinites with wide scope readings). I leave this as a matter for further investigation.

4.4.1.4. Names and Pronouns

Names and pronouns can only form non-restrictive L-trees. They map onto entities. They do not contain metavariables in the way characteristic of common nouns. From this follows the fact that it is not possible to derive restrictive L-trees for them. It is well known that names and pronouns cannot be modified by restrictive relative clauses, as (102) and (103) illustrate.

(102) a. *John who I met likes Jo.
b. John, who I met, likes Jo.
(103) a. *he who I met likes Jo.

b. *he, who I met, likes Jo.

(102) can only be given the non-restrictive reading (102b). (103) cannot even have this construal.

Both names and pronouns can, however, be used with discontinuous relatives, as (104) and (105) show.

(104) John arrived who I met.
(105) he's just walked in who I was telling you about.

This is to be predicted on an account which distinguishes the relative clause as input from the representations of content which may be derived from that; and where the restrictive/non-restrictive distinction is separated from the parenthetical/non-parenthetical distinction. The relative is processed as normal, building an L-tree. The wh- search procedure identifies the node of type (e) regardless of the fact that it has no internal structure. The L-tree derived is non-restrictive.

4.4.2. Syntax and Function

The problems that extraposed relatives provoke for transformational approaches were outlined in 4.2. and centre round the following questions:

i. are these relatives base-generated in their surface position or extraposed there from an underlying position adjacent to the noun?
ii. do they take as their antecedent the maximal projection of N or some subpart N'?
iii. what is the attachment point, where do they adjoin?

I will try to relate these to the concerns and approach of LDSNL. Here there is no distinction between different levels of representation. Structure is built dynamically according to the input; so surface structure order is reflected in the incrementality of tree growth. The final representation shows the semantic dependencies. To translate (ii), then, the answer to this can be either, depending

on the interpretation derived. Constituency tests reflect intuitions about the manner of combination, and are intimately bound up with processing issues.

The <LINK> operation allows a formal definition of the notions of restrictive and non-restrictive, which is distinct from whether or not the input constituent is parenthetical. Where the L-tree is a restrictive one, then this L-tree should not have the primary function, as in that case the function of the L-tree lies in identifying the entity. So what would the function of discontinuity be? Why not just have the relative clause adjacent to the noun phrase?

There may be considerations in terms of the overall structure - that it is just not possible to achieve adjacency, for example if there is other modifying material in between. It may be that the L-tree is additional material inserted for clarification, where otherwise there might be a failure of reference. Or it may be that this is a particular conjunction/co-ordination strategy where there is a degree of closeness between the two clauses that would not be communicated by use of 'and'. In any of these cases it is clear that the matter is not confined to syntactic considerations.

Non-restrictive modification by a discontinuous constituent is unsurprising. How to derive restrictive structure for a discontinuous modifier, where that is itself a fully tensed proposition, might seem more challenging. The <LINK> analysis solves this in the ways outlined below:

i. Compilation does not take place until 'all the chips are in', ie until all the premises have been admitted to the process, and there is some (intonational) indication that the (current) reasoning task is complete;
ii. there is no need to reconfigure the structure, as the linked database is in that sense an independent database; the variable is carried over without prejudice to any existing structure (and thus is true adjunction);
iii. that variable is now taken to have an existence of its own and is represented as such;
iv. just as with non-restrictive cases where there is intervening structure, the node to be carried over is sought earlier in the process. The operation is uniform.

4.5. Conclusion

The strength of the current approach is that the structures for discontinuous relative clauses can be derived by the application of a basic set of rules; all other information is derived solely from the lexical items, being information that these items contribute to the interpretation process anyway, and the interaction with context. This results in a much simpler model of 'grammar', where all and only the presented information is relevant for structure. At the same time it allows for a more specific approach to the integration of pragmatic factors, in terms of informational status and general reasoning.

The underdeterminacy displayed by discontinuous relatives is an example of the way in which pragmatics has to be brought in to the grammar, and how intimately grammar is bound up with processing. The properties of the relative are determined by the noun phrase, but not by preordained rule: all noun phrases built up from a common noun have a variable in them. This must rather have to do with the accessibility of this variable to other operations, and this is determined by what has already been done in the process.

Chapter Five

Discontinuous Constituency and Type (p) <LINK>: Extraposition and Adjunction

5.0. Introduction

In chapter three I introduced the distinction between trees of type (t) and trees of type (p) according to the type of predication, and I discussed the derivation of type (p) L-trees for 'reduced relatives'. In chapter four I gave an account of type (t) L-trees derived from discontinuous input, so-called 'extraposed relatives'. Here I investigate type (p) L-trees which are derived from discontinuous input. This involves predicates which have been analysed as extraposed or as adjunct. I examine what the distinction between restrictive and non-restrictive means with regard to type (p) L-trees and how this differs from 'full' relatives. L-trees of type (p) and type (t) differ in the way that they contribute to the process of interpretation, but this is related to their content rather than to substantive differences in the operation by which structure is built.

I note here that again there is the problem of underspecification. A single surface form may lead to a number of different types of interpretation. These structures are all in some sense modifiers, but what they modify may vary; hence in the sort of system proposed here, the structural operation which derives modification may differ.[1] This is particularly a problem in the case of prepositional phrases which may appear as adjuncts, arguments or extraposed modifiers. While the linguistic system underspecifies the eventual interpretation, the process must be able to show where this underspecification is located and how it is possible to derive the correct options.

My primary concern is <LINK> structures and their interpretation. In this chapter I explore how structure is built for L-trees of type (p) from discontinuous structure, and how this analysis ranges over all the basic

[1] See also the discussion in chapter three.

predicates. I will consider why these have been associated with some form of 'adverbial' interpretation. As is the case with discontinuous relatives, it will be seen that certain restrictions on acceptability are pragmatic. This is the case even though the data under discussion here might seem syntactically more integrated. I take this as confirming evidence for the general approach.

5.1. P-Predicates, L-trees and Discontinuous Input

In the discussion thus far L-trees of type (t) and L-trees of type (p) have not differed in their properties. When derived from discontinuous input, however, they do display clear differences. L-trees of type (t) derived from discontinuous input may be restrictive or non-restrictive. Discontinuous input of type (p) seems to give only non-restrictive L-trees. L-trees of type (t) do not differ in terms of their truth conditions according to whether they are derived from continuous or discontinuous input. L-trees of type (p) do. This is discussed in detail in 5.2.

Discontinuous p-predicates as a whole have not been analysed as a case of extraposition. What can be viewed as a single phenomenon from the perspective of dynamic structure building in the account developed here covers a range of data which have been treated as heterogeneous. In this section I review the areas of potential application of the type (p) L-tree, and delimit the area I wish to cover. This is restricted to cases of adjunction, where the basic predicate is entirely optional.

5.1.1. Secondary Predication Modifying the Subject

Trees of type (p) are derived from what I have termed the basic predicates, that is to say lexical items which can be used predicatively but which do not supply tense. In terms of being discontinuous, these can be split up according to the possible interpretations that they allow. Adjectives and participles, describing another property or action of the subject at the time of the main event, are shown in examples (1)-(5). I follow Rapoport (1991) in adopting the term 'adjunct predicate' to describe these, though without prejudice to the

possible structural configuration (cf also McNulty 1988). This is also neutral with respect to interpretation, unlike the term 'circumstantial' used in Roberts (1988).

(1) John drank his Campari sitting at the bar.
(2) Alice struggled off the tube hampered by the other passengers.
(3) a man left the pub blind drunk.
(4) the children ate their tea tired out by the excitement of the day.
(5) some of the guests managed to take food holding their glasses and their plates.

However, these cannot be understood as a *restriction* on the noun phrase. In (6a) 'drenched in sweat' describes a property of some man, presumably identified, which holds simultaneously with his coming into the locker room. In (6b) 'drenched in sweat' is used restrictively to identify the referent.

(6) a. the man came into the locker room drenched in sweat.
b. the man drenched in sweat came into the locker room.

This property contrasts with type (t) L-trees, where the discontinuous case can also have the restrictive interpretation. So both (6b) and (7a) can be paraphrased as (7b), but (6a) cannot.

(7) a. the man came into the locker room who was drenched in sweat.
b. the man who was drenched in sweat came into the locker room.

In contrast, it has been claimed that prepositional phrases are able to act as restrictors, in the same way that they can when adjacent to the noun phrase (cf Guéron 1980). (8) gives an example of this.

(8) a. a man arrived with green eyes.
b. a man with green eyes arrived.

For this reason, this particular construction has been taken to be extraposition, and in discussion of extraposition in the literature prepositional phrases are

generally included as a matter of course (see for example Guéron op. cit., Baltin 1984, Rochemont and Culicover 1990 and the references cited in chapter four).

My own intuition for this example is that here too the p-predicate is non-restrictive. It is certainly the case that there are examples where prepositional phrases may be 'circumstantial', as (9) and (10) show; these cases at least can be analysed as L-trees from discontinuous input.

(9) John showed up at the party with a hat on his head.
(10) the woman sped round the corner with a broad grin.

A complicating factor is that the prepositional phrase may be ambiguous between modifying the noun or modifying the action, and it can be difficult to separate these. Thus, in (11) it is unclear whether it is the event which takes place in the kitchen (in which case John would anyway be there); or whether it is John who is located in the kitchen (in which case the event will anyway take place there).

(11) John ate the noodles in the kitchen.

A further problem is the way in which prepositional phrases can combine with the semantics of the verb, relating a noun in some way to the action described by the verb.

(12) John wrote the letter with a pen.

In (12), it is clear that 'with a pen' describes something about the action, and is not a general property of John. In 5.3. I discuss the interpretive possibilities of circumstantials; in 5.5. I discuss the properties of prepositional phrases.

5.1.2. Secondary Predication Modifying the Object

Predicates associated with the object have also been termed adjunct predicates (Rapoport (1991), McNulty (1988)). As with the subject-oriented cases, these all

fall into the category of secondary predication.[2] The reason why subject modifying predicates provide a good test bed for my present concerns is because they provide a clear case of discontinuous input where the additional predication is optional, minimising the possibility of interference from the semantics of the verb. When the object is modified, it may be that this forms part of a different construction. Before I consider in detail the cases involving <LINK> and p-predication, I will review the different constructions involving the object of the main clause interacting with secondary predicates.

5.1.2.1. Object Depictives

(13) and (14) give examples of object depictives. What characterises these is that some predicate holds of the object noun phrase in the main clause, at the same time as the predication in the main verb, but that there is no necessary connection between the two predicates.

(13) Ceri served the beef sizzling.
(14) Isambard drank the milk heated.

Note the paradigmatic examples below. In (15) the adjective is modifying the subject and in (16) it is modifying the object. In the object case this example is not in fact discontinuous.

(15) John ate the meat naked.
(16) John ate the meat raw.

Moreover, these examples do not display the same behaviour as each other in terms of, for example, clefting.

(17) what John did crying/nude was leave the house
(18) *what John did raw was eat the meat

[2] For a recent overview of what data and debates this term covers in the generative paradigm, see Legendre (1997).

However, in both cases, in interpretive terms these are identical in that an additional predicate modifies some noun phrase in the main part of the sentence, resulting in an 'adverbial' type of interpretation.[3] They should therefore both be examples of type (p) L-trees.

5.1.2.2. Object Depictives and Reduced Relatives

The same surface form may be assigned a number of different interpretations. I start with the case where there is continuous input. Consider example (19).

(19) Isambard drank the milk heated to room temperature.

This is ambiguous between an interpretation where the predicate is being used to identify what milk is being drunk ie a reduced relative, and an interpretation where it provides some additional information about whatever it was that was drunk. This is exactly the distinction that was discussed in chapter four: the difference between restrictive and non-restrictive L-trees. (20) and (21) bring this out more clearly.

(20) John served the meat raw.
(21) John served the meat bought from the organic suppliers.

In (20) there is a clear preference for an interpretation as a non-restrictive, whereas in (21) there is a preference for the restrictive interpretation.

When these are discontinuous it becomes more difficult to derive the restrictive interpretation, as (22) and (23) show. (22) seems odd, as the preferred reading (in terms of meaning) is the restrictive one, but this is more difficult to obtain (in terms of processing). The particular predicate in this context does not work well as an adjunct.

(22) John tended all the soldiers yesterday wounded in the head.
(23) John served the meat yesterday bought from the organic suppliers.

[3] This is an intuitive and necessarily vague notion. I discuss this below and give a precise characterisation in chapter 6.

What is interesting about (23) is that in fact it does not have the same sort of interpretation as is generally associated with adjunct predicates. While this provides an additional predicate on the noun, this does not have the same sort of adverbial 'flavour' as the standard cases. That is to say, the predicates seem to be more distant; there is less of a relation between the two. (23) is not a suitable reply to the question (24).

(24) how did John serve the meat yesterday?

The explanation for this might be sought in relating this to tense, in that 'bought' is a past participle. However, there are arguments against this on two fronts. On the one hand, as (25) shows, it is possible to have a past participle which is 'adverbial'. This *is* a felicitous answer to (24).

(25) John served the meat yesterday burnt to a crisp.

On the other hand, the same non-restrictive, non-adverbial interpretation can be replicated in other cases:

(26) John fed the puppies yesterday yelping like mad things.

This is not a suitable answer to either (27) or (28).

(27) how did John feed the puppies yesterday?
(28) which puppies did John feed yesterday?

This may be preferred with the comma intonation associated with parentheticals. The reasons for this, and a (structural) definition of the term 'parenthetical', are given in chapter 6.

In summary, there are three possible types of interpretation for the object noun plus optional predication:

i. Noun phrase plus identifying restriction.

These are the reduced relative cases discussed in chapter three.

ii. Noun phrase plus non-restrictive L-tree with an adverbial interpretation.

iii. Noun phrase plus non-restrictive L-tree with interpretation as additional information.

These parallel the differences in extraposed relative clauses discussed in chapter four. The present framework provides the means to express these differences, as I discuss in detail below.

5.1.3. Secondary Predication and SCCR Constructions

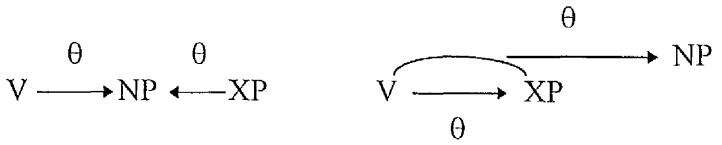
In the adjunct cases of object plus predicate, the predicate gives some other property of the noun, but this is unrelated to the main verb. In this regard they differ from those cases of secondary predication where there is an intrinsic semantic link between the verb and the secondary predicate. Rapoport (1991) makes the point that adjunct predicates should be distinguished from what she terms the SCCR (small clause, causative, resultative) cases where the verb may be said to select for some complement of the form noun plus predicate. (29) is a verb which subcategorises for a small clause; (30) is a perception verb example; (31) is a causative and (32) a resultative.

(29) I consider Sam a fool.
(30) John heard Dana singing.
(31) Terry made Sandra leave.
(32) I towelled my hair dry.

She characterises the difference in theta relations as shown in (33), (Rapoport 1991:162, her example (3)). In the case of the adjunct predicate both the verb and the predicate (separately) assign theta roles to the noun phrase. In SCCR constructions, the secondary predicate forms the complement of the verb.[4]

[4] These are discussed further in Rapoport (1993).

(33) Adjunct predicate SCCR



These differences between the two types of structure are motivated by a number of arguments, (cf Rapoport (1991), also Rapoport (1987) and Rothstein (1983)). Whether or not these might be amenable to a <LINK> analysis would have to be decided on a case by case basis. This is outside the scope of the present work, which is concerned with <LINK> structures only as adjuncts, not as complements. <LINK> structures may prove to be a fruitful analysis of resultatives, but this is a matter for further research.[5]

5.2. Discontinuous Input and Type (p) <LINK> Structures

5.2.1. Constructing L-trees of Type (p)

In the above discussion I have assumed that it is possible to derive L-trees for P-predicates from discontinuous input. I now address the details of this. (34) gives the general rule for deriving L-trees of type (p) stated in chapter three. This was developed in the context of continuous input, more specifically to account for the case of reduced relatives.

(34) The <LINK> Rule for P-predication

$$\frac{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)]}{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)], \, [_L \bullet \, Ty(p); \, [_* \, Fo(\alpha), \, Ty(e))]]}$$

[5] As (i) illustrates, in resultatives the action described in the main verb leads to the state of affairs described by the secondary predicate.

(i) Melanie painted the kitchen orange.

The extent to which the verb should be analysed as subcategorising for the predicate is a controversial issue. Goldberg (1995) gives a detailed review of accounts of resultatives across frameworks, and proposes an analysis in Construction Grammar.

In order to avoid any requirement for identity between the node from where the <LINK> operation is launched and the node shared between the two trees, and in order to ensure that this rule may operate at any node location, I modify this slightly to the form given in (35).

(35) The <LINK> Rule for P-Predication (revised)

$$\frac{[_m \ldots [_n \, Fo(\alpha), Ty(e)] .. \, [_o \, Ty(y)] \ldots \bullet \, Ty(t)]}{[_m \ldots [_n \, Fo(\alpha), Ty(e)] .. \, [_o \, Ty(y)] \ldots \bullet \, Ty(t)], \, [_L \bullet \, Ty(p); \, [_* \, Fo(\alpha), \, Ty(e))] \,]}$$

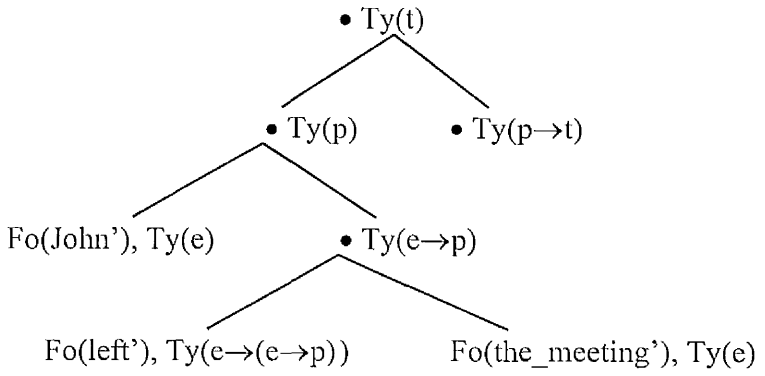
This rule states that at any location, tree node (o), which is annotated with a type description, and where at least one node is annotated with a description of type (e), it is possible to launch the <LINK> operation. The effect of this operation is to build an L-tree annotated with the overall requirement type (p), which also has imposed on it the necessity that there be some node held in common with the main tree. There is no effect on any tree relations established in the main tree. This characterisation of the rule allows for type (p) L-trees to be built for discontinuous input; the location of the shared node in the main tree (ie the identity of the shared node) is not specified, and hence this may be either the subject or the object. Note that this rule will still work for cases where there is continuous input, the reduced relatives.

To illustrate the working of this rule in a case of discontinuous input, I examine the process of structure building for (36)

(36) John left the meeting angry.

This proceeds in the standard way, outlined in previous chapters, to give the derivation in (37) at the point at which the lexical input 'the meeting' has been scanned.

(37)



I suppress the internal structure of the object e-task here, as compilation has already occurred.

At this point the <LINK> rule can apply to give the representation shown in (38).

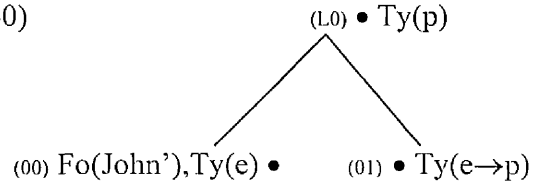
(38) (L0) • Ty(p); [$_*$ Fo(α), Ty(e))]

In this example, assume that the node that is shared between the two trees will be 'John'. The rule of Introduction allows this to be specified as sub-goals at child nodes as shown in (39).

(39) (L0) • Ty(p), [<d> • Ty(e)], [<d> • Ty(e→p)]; [$_*$ Fo(John'), Ty(e))]

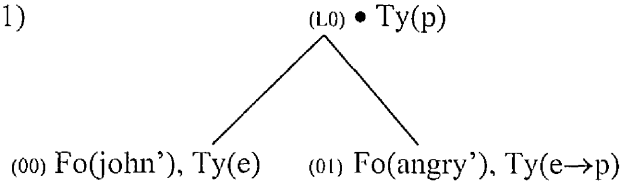
Pointer movement to tree node (00) is induced by application of the subject rule, and so the underlocation of the shared node can be resolved, and this is incorporated as the subject. Completion returns the pointer to node (L0) and Prediction allows node (01) to be built. This gives the structure in (40).

(40)



The next lexical premise to be scanned is 'angry', which duly annotates this predicate node. The final representation of the L-tree is (41).

(41)



The overall interpretation for the utterance (36) is given in (42), where a distinction is drawn between a tensed predication and a non-tensed one.[6]

(42) Fo(left'(the_meeting')(john')), Ty(t) & Fo(angry'(john')), Ty(p))

This account raises two related issues: how should the shared node be derived, and should it be the predicate itself ('angry' in this example) which induces operation of the <LINK> rule?

The process of building the L-tree can be broken down into a number of steps. The <LINK> rule states that there has to be a shared node, and moreover that this node has to be of type (e), but does not specify what this node actually is. Unlike the cases of continuous input originally discussed in connection with <LINK>, there does not have to be identity between the node from which the <LINK> operation is launched and the node carried over into the L-tree. In the case at hand nothing pre-determines the identity of the shared node. So does this node have to be identified before construction of the L-tree can proceed? Consider the examples below.

(43) a. Tony eats vegetables raw.
b. Tony eats vegetables naked.

In (43) it is the meaning of the predicate in the L-tree which determines whether the adjective should apply to the subject or the object. If this has to be

[6] What this actually means in detail (in semantic and pragmatic terms) is explained in chapter seven.

considered before the shared node can be identified, then the representation of the shared node in the L-tree should be characterised as a place-holding metavariable, of type (e), as in the case of wh- words. The identity of this is not predetermined, but has to match with the description of some node of type (e) in the main tree. Adopting this anaphoric characterisation provides a natural account of examples such as (44).

(44) Tony painted the man naked.

Either the man was naked and that is how Tony painted him, or Tony himself was naked during the painting. Here the underspecification cannot be resolved from the lexical content; there is a genuine ambiguity which can only be resolved by reference to wider contextual considerations.

The recasting of the <LINK> operation to cover cases from discontinuous input raises the question of what invokes its operation. In those cases of type (t) discussed in the previous chapter, this was associated with the relativiser. In the current case, however, there is no relativiser. Should the <LINK> rule be invoked in order that structure can be derived for the adjective, or is it the case that some structure building process is started for the adjective and then the <LINK> operation will guarantee a well-formed structure overall by creating the connection of this to the main tree?

The way that the lexical rules have been set up in the system, it has to be the case that there is some trigger of a requirement of type (e→p) in order for the action associated with the adjective to take place. However, if the requirement of type (t) is available by axiom, then given that deriving structure for natural language is a goal oriented process, there seems to be no reason why this should not also be the case for type (p).[7] The rules of Introduction and Prediction allow this to be broken down into the sub-goals of a requirement for type (e) and a

[7] The imposition of some temporal reference may be a general inferential phenomenon, rather than a linguistic necessity. Thus Hebrew supplies tense in the past and future but not the present. Tense is in any case modelled as an anaphoric device in LDSNL (see Gabbay (1996), Perrett (in prep.)). While it may be necessary for a proposition to acquire some form of temporal indexing in order to be evaluated, this is not necessarily a pre-requisite of deriving linguistic structure.

requirement for type (e→p). Given the correct movement of the pointer,[8] the adjective could be incorporated as the predicate, imposing the requirement for a subject. This would be supplied as a node shared with the main tree, guaranteeing a well-formed <LINK> structure.

Such an approach has intuitive appeal in that it goes further in incorporating the insight that structure building is lexically driven. Furthermore, developing this could lead to a more precise description of the circumstances in which the <LINK> rule can be invoked. However, I leave this route unexplored for the present. An account in these terms would result in undue formal complication in reformulating the rules. The present set-up, retaining <LINK> as a general processing option, allows for the same process of structure building for predicates across different tree environments. Moreover the process I have given correctly restricts the shared node to being of type (e) and in location (L00) (ie the subject position).

5.2.2. Type (p) L-trees Compared to Type (t) L-trees

I am proposing an analysis which provides a unified structure building operation for extraposition and adjunct predicates. According to general syntactic analyses, there should be no reason to give the same analyses to these sets of data in that they have distinct forms of interpretation and display different properties.[9] In the terms of the present framework this falls out as a consequence of general considerations. The process is driven by the requirements of the predicate itself, and the dynamics of the system. This allows a more uniform and hence simpler account. Reasons for the differences in interpretation lie elsewhere in the general properties of the predicates involved.

One of the problems that I have to deal with is the way that apparent syntactic characteristics can be dealt with in the present approach. Much of the work in LDSNL up till now has concentrated on areas where syntactic facts are derived directly from the dynamics of the system. In the present case it is not

---

[8] For example by utilising the Rule for Predicate Derival given in 3.6.

[9] In the case of prepositions the interpretation associated with either may be available, as is discussed in 5.5.

obvious that this is true in the same way. What can lead to differences in syntactic properties?

As regards the differences between type (p) L-trees and type(t) L-trees when derived form discontinuous input, there are two areas where these may differ:

i. in terms of the input to the L-tree from the predicate itself (ie whether or not this supplies tense), and hence the end representation;
ii. in terms of the relative location of the processing tasks.

The role of the wh- relativiser is less central, though it does have some bearing inasmuch as it impacts on the importance of (ii). The wh- element affects the process of arriving at the structure ultimately derived, but does not have any representation in that structure; its role is procedural: it does not in itself contain conceptual material which has to be realised in the final structure.[10] Rather it instantiates a search and induces a copy procedure, creating a copy in the L-tree of some node description from the main tree. The fact that it launches the <LINK> operation is also not immediately relevant; this is a freely available option in the type (p) cases. The area where it does make a difference is in the accessibility of the referent. I assume, following Ariel (1990), that a zero form[11] indicates higher accessibility of a referent than a pronoun. Specifically, the wh-pronoun indicates that the shared node may be found at a greater distance, though not necessarily that it has to be. This makes it more likely in the case of t-predicates that a restrictive L-tree can be built, as the metavariable lies at a greater distance than the e-node. The zero form of the 'reduced' relative indicates that the shared node has to be of high accessibility, and so in the case of the type (p) L-tree the antecedent is likely to be the more accessible e-node, giving a non-restrictive L-tree. I return to this in 5.2.4.

What is the syntactic basis of distinguishing extraposition and adjunct predication? Kroch and Joshi (1987) note that with full relative clauses, the

---

[10] As a pronoun it is effectively a resumptive. In LDSNL the final structure reflects only the semantic relations. Resumptives and gaps lead to the same representation.
[11] In the terms of the present framework a zero form means a node which is derived, but not by a lexically driven operation. Therefore there is no corresponding 'surface' form.

extraposed and non-extraposed display similar properties, but there are differences between an adjunct predicate modifying the noun and modifying postverbally. I consider these in terms of the analysis I have proposed.

L-trees of type (p) have different truth conditions according to whether they are derived from continuous or discontinuous input.

(45) the people left the theatre delighted by the performance.
(46) the people delighted by the performance left the theatre.

In (45) there is a set of people who have left the theatre, of whom the further fact is true that they were delighted by the performance. In (46) the predicate restricts the set of people such that they were delighted by the performance, and it is these people who left the theatre. Kroch and Joshi characterise (45) as having an adverbial interpretation; (46) is restrictive.

In contrast to this is the meaning exhibited by the type (t) L-trees. Here there is no necessary difference in truth conditions.

(47) the people left the theatre who were delighted by the performance.
(48) the people who were delighted by the performance left the theatre.

The difference in terms of the structures derived is that between restrictive and non-restrictive L-trees. That is to say: is the shared node the metavariable from the common noun or the completed e-node? It was noted in chapter four that type (t) L-trees from discontinuous input can be either non-restrictive or restrictive. As the interpretation of (46) shows, when there is adjacency, that is to say when the input is continuous, the type (p) L-tree can be restrictive. This leaves the question of why it should be that the restrictive L-tree cannot be derived in the discontinuous case for the p-predicate.

Kroch and Joshi make a further distinction between extraposition and adjunct predicates in terms of their interaction with pronominals. In the case of the type (p) L-trees, these are ungrammatical when adjacent, but grammatical when discontinuous.

(49) he left the room happy.
(50) *he happy left the room.

In contrast, with type (t) L-trees, it is claimed that neither of these is acceptable.

(51) *he who is happy left the room.
(52) *he left the room who is happy.

Again this can be explained in terms of the difference between restrictive and non-restrictive. A restrictive L-tree is not possible with a pronoun, which lacks any available metavariable, and itself identifies a referent without the possibility of further restriction. In the infelicitous cases above, these should be restrictive, but cannot be. The type (p) L-tree when discontinuous, (49), is non-restrictive and is therefore acceptable.

This is brought out in the examples below, which illustrate that it is also possible to see this difference with type (t) L-trees. 'That' forces the restrictive construal, and so (53) is ghastly. However, 'who' allows a non-restrictive interpretation and (54) is acceptable.

(53) *he turned up that I was telling you about.
(54) he turned up who I was telling you about.

The same facts are displayed with proper names, which like pronouns do not allow the derivation of a restrictive L-tree. In the absence of parenthetical intonation, (55) is alright, whereas (56) is unacceptable.

(55) ?John left the room who is happy.
(56) *John who is happy left the room.

What the above discussion brings out is the point that type (p) L-trees cannot be restrictive.

5.2.3. Type (p) L-trees, Quantification and Truth Conditions

Type (p) L-trees constructed from discontinuous input can only be non-restrictive. This conclusion is reinforced by considering them with quantified phrases; this also sheds light on how these are affected by the processing domain.

In the above discussion the examples consisted solely of definite noun phrases. In the case of definites with L-trees of type (t), these can be either restrictive or non-restrictive. With type (p) these have to be non-restrictive. However, in chapter four the different types of noun phrase were seen to affect the overall interpretation of the utterance with the <LINK> structure. What effect do these have on these adjunct predicates? In all cases these are stubbornly non-restrictive.

Quantified phrases always result in the restrictive reading when associated with an L-tree of type (t).

(57) every person went home who was sick.

In the case of type (p), however, these are not restrictive.

(58) a. every girl disappointed by her exam score entered the classroom.
b. every girl entered the classroom disappointed by her exam score.

In (58a) the predicate is restrictive and delineates the set referred to by the noun phrase. In (58b) the predicate gives an additional property of a set of individuals that has already been established. Examples (59)-(62) provide further illustration of this. In all of these, in the (a) examples the basic predicate is restrictive of the interpretation of the noun phrase; in the (b) examples the basic predicate specifies a further property or action of the subject.

(59) a. most people invited arrived.
b. most people arrived invited.
(60) a. that man speaking on his mobile phone drove his car.
b. that man drove his car speaking on his mobile phone.
(61) a. a girl reading a book sat down at the table.

b. a girl sat down at the table reading a book.

(62) a. a child red in the face stumbled across the finishing line.
b. a child stumbled across the finishing line red in the face.

How should the truth conditions of these be characterised? Taking the simple case with a proper name, it is straightforward to specify the truth conditions (ignoring tense for the moment). Thus (63) maps onto (64).

(63) John sat down exhausted.
(64) ((sat_down)(john) & (exhausted)(john))

The result is two conjoined predications, which is exactly what the <LINK> operation would derive. This corresponds to intuitive feelings about what predication is involved in (63), and to the logical implications that it has, namely that John sat down and that John was exhausted.

However, bringing quantification into the picture complicates matters considerably. The correct characterisation of the truth conditions of (65) is given in (66), (again leaving aside tense).[12] It may be that there are additional interpretive effects associated with the type (p) L-tree. Nonetheless, the truth conditions are such that every girl entered the classroom, and every girl was disappointed by her exam score, and that the same girls are being referred to.

(65) every girl entered the classroom disappointed by her exam score.
(66) $\forall x(\text{girl}(x) \rightarrow (\text{entered...}(x) \ \& \ \text{disappointed...}(x)))$

The quantifier binding is demonstrated by the possessive: it has to be the case that each girl is disappointed by her own exam score, rather than by each other's.[13]

As a point of comparison, the truth conditions for the restrictive L-tree derived from continuous input (67) is (68).

[12] What supplies tense is crucial in terms of the inferential effects; but the temporal evaluation is not relevant to the discussion here. I assume that tense is marked on the utterance as a whole. For an examination of temporal dependencies between clauses see Perrett (in prep.).
[13] Though it may be that 'her' refers to a different individual altogether.

(67) every girl disappointed by her exam score entered the classroom.
(68) $\forall(x)((\text{girl}(x) \ \& \ \text{disappointed...}(x)) \rightarrow \text{entered...}(x))$

What is interesting about (65) is that this involves a case where the L-tree is non-restrictive, but where the quantifier in the e-node nonetheless apparently binds the L-tree. How does this come about?

To explain this requires a brief discussion of the way that quantification is processed in LDSNL.[14] Recall how compilation proceeds for a quantified expression. Combining a quantifier (type(cn→e)) and a common noun node (type (cn)) leads to the creation of a term in Hilbert's epsilon calculus (Hilbert and Bernays, 1939). Taking these as the basic terms but incorporating the sense in which natural language underspecifies such terms, these are said to consist of a triple: the operator, a metavariable and the restrictor (the binding context). Existential quantification uses the epsilon operator, universal quantification is handled by the tau operator. (69) gives the general form of these; (70) shows the term which, through compilation, is constructed for 'every man.'

(69) a. Existential (ε, x, R)
b. Universal (τ, x, R)
where x is a variable and R is the restrictor.

(70) (τ, x, Man)

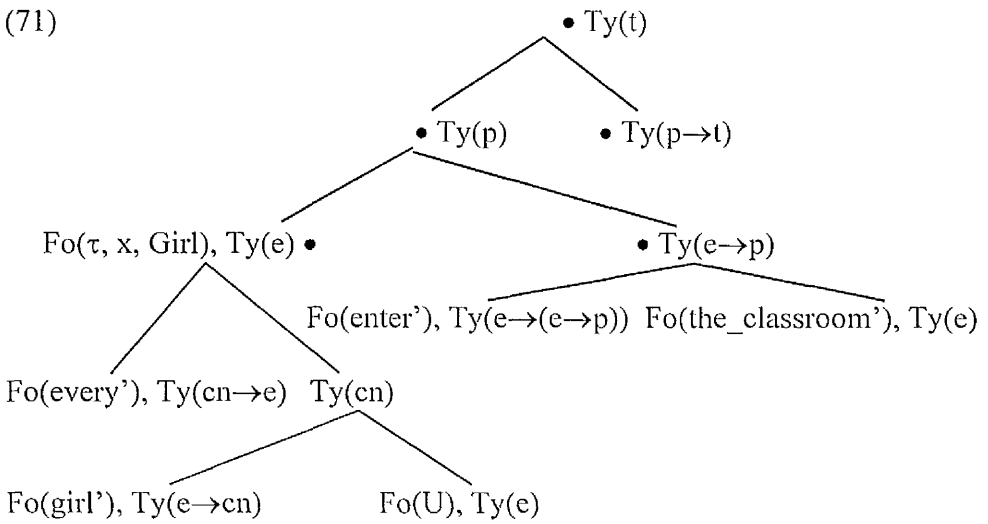
The representation derived by compiling the internal structure of the e-node acts as an arbitrary name in terms of the subsequent development of the tree structure. This effectively provides a means of quantifier storage.[15] The arbitrary name which the term represents stands in for the quantifier until such point as the quantifiers can be reintroduced and the effects of dependencies computed. This does not happen until the whole tree has been compiled and the pointer has

[14] This is outlined in detail in Kempson, Meyer-Viol & Gabbay (fthcmng), chapter seven.
[15] The process of determining quantifier scope is characterised as an anaphoric-like choice, generally made on-line. The quantifying expression creates a dependency predicate as an annotation on the root node. The second argument to this predicate is a slot for the argument on which the quantifying expression being processed is dependent. This slot is filled as a process of enrichment as the emergent tree is established. The dependency choice thus made fixes the way the quantifiers are later resolved at the propositional level. The effects of that choice on quantifier scope are then computed at the root node, type (t).

returned to the tense node. In this respect, despite the fact that dependencies are chosen on-line, quantifier scope itself remains not fully defined until the tree is completed, obviating any need to reconfigure the tree.

This approach to quantification provides a natural account of the present case. The process of structure building for (65) takes place as standard for a regular main clause. (71) shows the state of the structure at the point immediately prior to the building of the <LINK> tree.

(71)

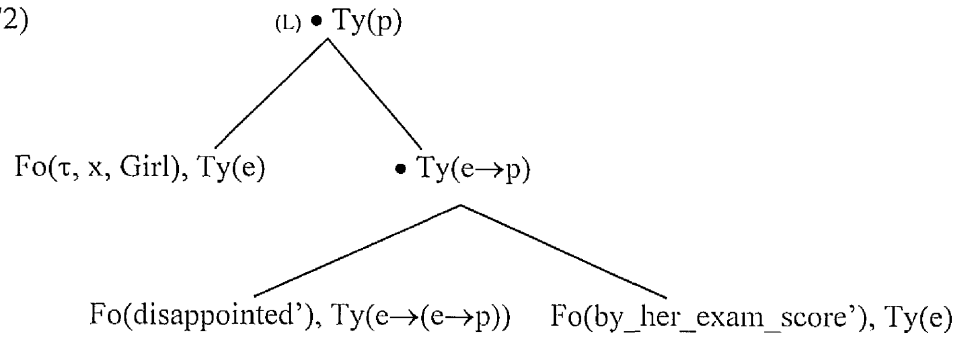


Note that at this stage there are no further requirements. As the above tree shows, compilation has occurred within the e-node, resulting in the creation of a tau term. However, overall compilation has not yet taken place.

At this stage there is more input, namely, 'disappointed by her exam score'. The <LINK> operation applies, which introduces the necessity that some node be identified which is of type (e).[16] What is available to be carried across into the L-tree is the tau term which forms the description on the e-node 'every girl'. This is duly carried across. 'disappointed...' is scanned, and the result is the building of the L-tree structure as shown in (72).

[16] 'The classroom' can be ruled out on pragmatic grounds.

(72)



Note then that the e-node which is shared between the two trees has itself been compiled; its internal structure is not available to be reconfigured by any processes in the L-tree. The tau term that is carried over is effectively a name; as far as its role in the L-tree is concerned, it has no internal structure. When compilation of the L-tree occurs, the tau term still cannot be unpacked, as there is no tense to allow quantifier relations to be established. It is therefore dependent on the main tree for its interpretation, and cannot be separately evaluated. It is only when compilation of the main tree has been completed that scope dependencies are computed and the quantified representation can be derived.[17] Note also that the <LINK> rule requires strict identity of nodes shared between trees so there is no possibility of establishing separate interpretations for the two occurrences of the tau term. Hence the overall truth-conditional interpretation derived is (73), repeated from above.

(73) $\forall x(\text{girl}(x) \rightarrow (\text{entered...}(x) \ \& \ \text{disappointed...}(x)))$

The processing of the quantification is parallel to the processing of the tree structures overall. When it comes to compilation of the whole, the <LINK> tree for a p-predicate is a separate part. It is not marked with tense, and so is a minor premise which cannot constitute the main force of the utterance. While it does form a separate constituent as a proposition, it cannot be separately evaluated with regard to truth, as at no level does it form a separate assertion. So it is not

[17] For technical details of how this proceeds, see Kempson, Meyer-Viol & Gabbay (fthcmng), chapter seven.

an independent vehicle for inference: it is dependent on the main tree and remains within the same overall t-task. Compilation occurs for each of the trees; but the interpretation of the L-tree is within the same overall goal of the utterance.

More widely, there is the problem of how to capture the communicative intent of the utterance in terms of the way that two (or more) propositions have been juxtaposed. What information can the semantic content, defined as truth conditions, capture?

Bracketing within a flat representation has to indicate the 'processing hierarchies', but these can be either scope relations or order of application of predicates (which itself may be determined or underdetermined by the input form). So for the example (74), universal quantification first binds the restrictive terms to identify the basic entity, but then binds an additional occurrence of the variable with the main verb.

(74) every boy upset by the news burst into tears.
(75) $\forall(x)((boy(x) \ \& \ upset...(x)) \rightarrow burst...(x))$

The general problem with defining truth for sentential units is that although this appears to be straightforward for simple propositions, as soon as there is more than a basic predication there arises the problem of interrelating the parts to the whole, and how this relates to the actual communicative intent. Indeed, even in the case of simple sentences, the truth-conditional representation does not indicate the mode of presentation of the information, and hence its actual cognitive and communicative effects.

Consider the basic case presented in (76). (77) may be the correct specification of the truth conditions, but this says nothing of the way this has been derived. It does not reflect *linguistic* meaning in anyway.

(76) a man arrived.
(77) $\exists x(man(x) \ \& \ arrive\ (x))$

In approaches such as Discourse Representation Theory (DRT) (Kamp 1981, Heim 1981) a discourse representation schema reflects a richer notion of content. [18] The sorts of representation developed in LDSNL differ in that they seek to model the *process* of structure building step by step: it is directly derived from linguistic input with no intervening level of representation or set of mapping operations. Truth conditions can be directly defined over these structures, but semantic interpretation is directly bound up with the structure building process. Characterising the way that linguistic content is directed towards this process is a central concern of the present approach.

The LDSNL perspective is rooted in an approach to language which is resolutely cognitive and located in a very specific view of how language processing and cognitive processes operate. The idea of incremental tree growth effected by lexical input is central to the concept of interpretation and, in this sense, meaning. The notion of linguistic meaning, semantics in its broadest construal, is intimately tied up with procedures. The focus is not on what the eventual interpretation is, nor on simply incorporating a more structured notion of content; rather, the point is to indicate *how* this interpretation comes about.

What the representation of truth conditional meaning itself fails to reflect are considerations about the linguistic meaning: how the sentence structures the meaning, as this is considered a syntactic problem. In the case of secondary predicates under discussion, there are two problems for approaches with a strict divide between syntactic and semantic levels of representation. On the one hand, there is the problem of why different lexical categories should be associated with some particular adjunction structure; on the other hand, there is the problem of how they combine as predicates to achieve interpretation.

Characteristic of the present approach is the intimate connection between all aspects of the process, and the way that this case is assigned structure according to general principles of the system. An additional predicate enters the structure building process; for a well-formed structure to be derived, the <LINK> operation has to apply. The subcategorisation requirements of the predicate require a subject. The predicate lacks tense, so the tree derived has to play a

[18] See also Kamp and Reyle (1993).

minor role in the overall structure of the utterance, hence the characterisation as 'adverbial'. All of this derives from the general process of developing structure.

Thus far, the only semantics I have discussed for the <LINK> operation has been conjunction. Conjunction is not straightforward in the present case as the respective trees are of different types. It is therefore not the case that these involve straightforward conjunction of two equal parts. However, truth conditional conjunction is the final result of the <LINK> procedure; that is, both propositions have to be true simultaneously.

The crucial point about the type (p) L-trees is that they are not specified with tense. In these cases they have to be dependent clauses, and so may be characterised with some justification as secondary predications. What seem to be the implications of these depend on the particular predicate and how it is construed in context. In section 5.3. I investigate how interpretation is affected by this configuration. First I consider why it is the case that only non-restrictive L-trees can be derived for P-predicates when the input is discontinuous.

5.2.4. Non-restrictiveness and Identifying the Shared Node

What does it mean to draw the generalisation that when p-predicate input is discontinuous it is only possible to have a non-restrictive L-tree? To state this in different terms, in these circumstances the p-predicate can only have as the shared node in an L-tree a description resulting from an e-task. It cannot have the metavariable supplied by a common noun.

How should this property be captured in the system? It could be specified technically by imposing a restriction such that the e-node shared between the two trees has to have some denotational content. What this means is that there has to be some formula information as part of the description; this is something which does not hold of the metavariable. However, this restriction on the nature of the shared node is not an intrinsic property of p-predication. In the case of continuous input the basic predicates form perfectly good restrictive <LINK> structures. Moreover, it is not a general characteristic of the system that reference is made to denotational content in defining structural operations. In general, conditions requiring identity of representation require exactly that; the

system is defined syntactically and not with respect to denotational content. Therefore any specification of the type of node to be transferred cannot simply be stated as part of the <LINK> rule.

This would have to be made a context sensitive feature of the <LINK> operation, ie that a <LINK> rule triggered in the particular context would have this restriction written into it. Context sensitive rules are an integral part of the structure building system developed. Indeed, given that structural development is achieved by rewrite rules, all rules are in a sense context sensitive, triggered by a particular environment. However, context-sensitive rules as a means of reflecting underspecification have been associated specifically with lexical items. My aim is to keep the <LINK> strategy as general as possible. The general problem is why discontinuous p-predicates lead only to a non-restrictive interpretation: this is sensitive to a specific processing environment, but should be derivable from more general properties of the process, rather than requiring stipulation by rule.

Ariel (1990) develops a theory of Accessibility where different types of nominal may indicate different levels of accessibility of their referent. According to her analysis, a zero form marks extremely high accessibility of some antecedent, while a pronoun indicates that the object to which it refers is fairly accessible. From this perspective, it is unsurprising that an L-tree with a relative pronoun is able to be restrictive regardless of whether or not this is a case of continuous or discontinuous input. The way this relative pronoun is characterised in chapter four is as launching a search procedure. The wh-pronoun seeks some node description outside of its own processing domain (ie outside the L-tree) with which it can identify. The successful accomplishment of this results in a node description being shared between the two trees such that a well-formed <LINK> structure has been derived.

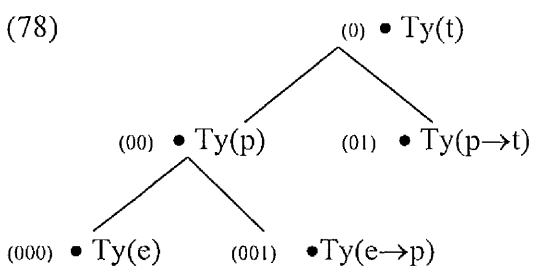
In the case of L-trees of type (p), there is no lexical item to launch a search procedure. What induces the requirement of some shared node content is the <LINK> rule itself. In this case, therefore, the node which is shared has to be much more accessible. What affects degrees of accessibility? The present approach is concerned with processing, so two lines of investigation present

themselves: activation and distance. By the idea of degrees of activation I mean the processing state of some mental referent, where that object is characterisable as a cognitive entity. The more highly activated a referent is, the more accessible it is, where factors such as recency of use, ease of construction, and processing status influence the degree of activation. This property does correlate with discourse and linguistic notions such as topic, where the topic is maintained as a highly activated entity; for example, Ariel (1994) examines the interaction of discourse topic and types of pronominal reference. For the present, however, I confine myself to linguistic factors definable in the present framework and how these affect some putative notion of distance.

5.2.4.1. Processing, Accessibility and Tree Structure

LDSNL as presented here uses semantic trees to indicate the process of structure building and the mode of combination. This gives a precise measure of degrees of distance, where these nodes indicate not lexical items or the heads of functional projections, but premises and loci of combination. These nodes are labelled according to the task specific to each and the overall tree relations indicate overriding processing tasks. Consider the tree outline given in (78).

(78)



Here the overall goal which characterises the tree is the requirement of type (t). Tree node (000) can be characterised in terms of task states in the following way: it is itself an e-task. It is located within a p-task. That p-task is part of a t-task. This derives from the parent/child relations, and reflects the way in which premises combine in the process of compilation. So for example, a p-task cannot be completed until the tasks of its sub-parts have been completed. Exactly what constitutes any difference in the behaviour and status of a singular type task

such as the p-task at tree node (00) and a composite task such as the e→p task at tree node (001), I leave as an open question for the moment. The reason why this might be a significant distinction is that in the case of a singular type what is derived is a completed object; whereas in a composite task, this in itself specifies the requirement for another object. In what follows I am only concerned with tasks that result in completed objects.

Some characterisation has to be made in terms of the relevant task state in which the operation is taking place: ie what the particular goal is at the point at which the <LINK> operation is launched. There are two factors at play here. The first is the overriding subgoal at the point at which the <LINK> operation is launched, ie what constitutes the task domain. The second is the actual position of the pointer in the tree description, ie the specific node location. With predicates of type (p), the L-tree remains within the overall inferential domain of the utterance, and hence within the same overall t-task. The importance of this is considered below. The local task within which the <LINK> operation is launched is to derive a proposition of type (p). The search for the node to be shared is limited to this task domain. The position of the pointer may vary (but will be somewhere within that domain).

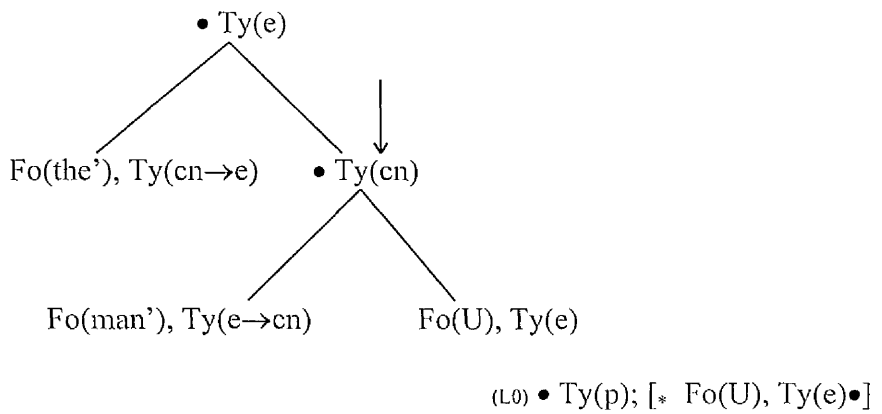
5.2.4.2. Accessibility and the Pointer

The pointer indicates where the process is in terms of tree development. Any characterisation of accessibility should therefore be made in relation to the pointer. The node which is identified as shared between the L-tree and the main tree has to be highly accessible; what this means is that not only does it have to be within the same processing domain, but it can be only one node away from wherever the pointer location is; that is, it has to be located at a child of the node at which the pointer is. Pointer movement is constrained by general considerations of the process, and reflects how that process unfolds. The pointer cannot be moved freely. This explains why the cases with discontinuous input can never be restrictive. The metavariable is never accessible in terms of nodes.

In the case of continuous input,[19] the overall task domain is the e-task. When the <LINK> operation is launched, the pointer is at the cn-task, as the arrow shows. the L-tree requires some node description of type (e), which has to be located at a child node. Only the metavariable is of the requisite type. This is duly identified as the shared node, and carried over into the L-tree. This is shown in (80).

(79) the man angry...

(80)



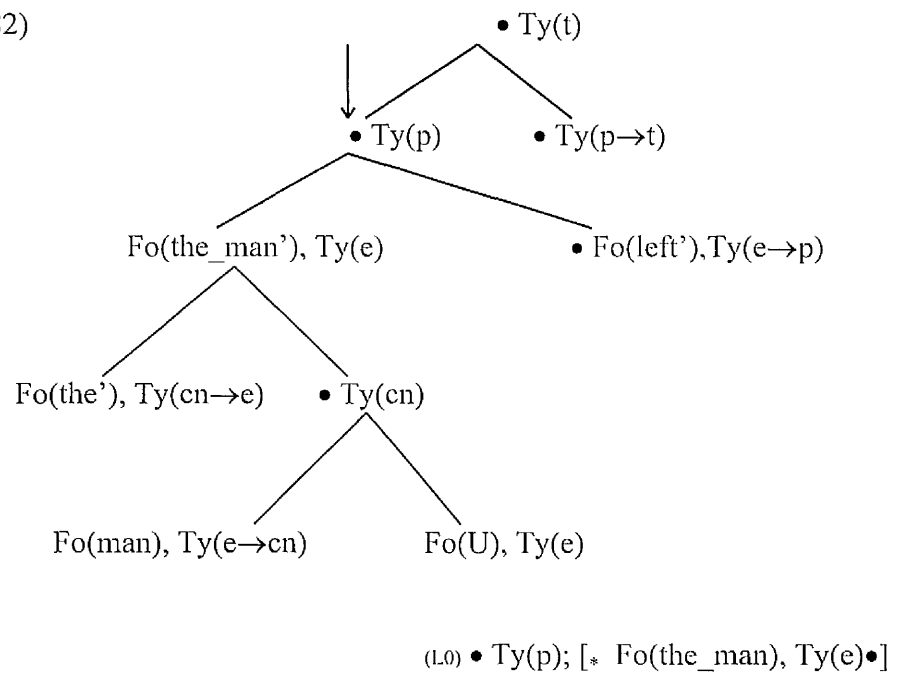
In the case of discontinuous input, the task domain from which the <LINK> operation is launched is the 'p' task. The subject is at a greater distance, but crucially it lies within the *same* processing task. That task is type (p). In this sense it is still sufficiently accessible, and can be the shared element between the two trees (provided that the pointer is in the right position). In the example given in (81), after the verb has been scanned the pointer returns to the node of requirement type (p), as (82) shows. From here there is a child node of type (e), with the formula description "the_man'". In fact, that this description holds at a daughter is annotated on the node of type (p), due to the earlier step of Completion.[20] The e-node, then, is identified as the shared node, carried over into the L-tree.

[19] Where by 'continuous' I mean there is no intervening lexical material and no intonational break.

[20] Following standard practice, this is not shown in the tree representation in (82).

(81) the man left angry

(82)



This is why it is impossible for these type (p) L-trees ever to be restrictive. Accessibility in this case has been defined in terms of relation to the pointer location: that the node to be identified has to be a child of the node at which the pointer is located. In order for this to be true of the metavariable the pointer would have to at the cn task node. Nothing can induce the pointer to move there.

This is in contrast to cases where there is an object but the subject is still identified as the shared node.

(83) Tony ate the chicken naked.
(84) Tony ate the chicken roasted.

In this case, the object might be the shared node, as (84) shows. Identification of the object as the shared node is available when the pointer is at the node of type (e→p). However, this interpretation for (83) can be ruled out on pragmatic grounds.[21] What happens subsequently in terms of tree development? The verb and object combine, and compilation returns the pointer to the p-node. From this

[21] Note that according to Relevance Theory, the acceptable interpretation is not the one that involves least processing effort, but the one that achieves adequate effects for least processing effort (cf Sperber and Wilson 1986/95).

position the subject is now accessible, as it is in a child relation to this node. Thus, the subject can be instantiated as the shared node.

In (5.1.2.2.) I discussed the possible ambiguity when a p-predicate follows the object noun. It can be understood either as a reduced (restrictive) relative, as in (85), or as a depictive, as in (86).

(85) Alice bought the chicken raised free range.
(86) Alice bought the chicken ready cooked.

In the restrictive case, the shared node is identified from the cn-task and is the metavariable. In the depictive the internal structure of the e-task is compiled, and it is the e-node itself which is carried over.

The way in which accessibility has been defined above sheds light on an apparent anomaly in the way that object depictives behave with regard to discontinuity. Consider the data in (87)-(90).

(87) Noa ate the meat raw.
(88) ??Noa ate the meat quickly raw.
(89) ??Noa ate the meat in the kitchen raw.
(90) ??Noa ate the meat with chopsticks raw.

As soon as any material intervenes between the object and the adjunct predicate, acceptability decreases. The same effect is not observed with subject depictives, as (91)-94) show.

(91) Noa ate the meat naked.
(92) Noa ate the meat quickly naked.
(93) Noa ate the meat in the kitchen naked.
(94) Noa ate the meat with chopsticks naked.

In the object case, for the main predication to be modified, the pointer has to move up to the p-node. This movement is achieved by compilation of the verb with the object. In this case, nothing can move the pointer back to a node location where the object is located one node below. This is in contrast to the

subject, where this e-node is a child to the p-node. Note also that intervening material is fine when in the e-star domain, as (95) illustrates.

(95) Sam gave the parcel to Joan unopened.

In this case the indirect object is incorporated into the verb as an argument, and as compilation moves back up the tree, the direct object is accessible.

When the <LINK> operation is launched, the process is within a certain task domain. The shared node has to be found within the same task domain. So when the L-tree is within the same t-task, then the node shared between the two trees has to come from within that processing task. The node identified as the shared node has to be highly accessible, where this is defined as being in a child relation to any node at which the pointer is located. The L-tree associated with the p-predicate constitutes a localised processing phenomenon. I now consider the effects that this has on interpretation.

5.3. Interpretation

Adjunct predicates have been identified as 'adverbial'; as secondary predicates they do not carry the main force of the utterance. Rather, their role is to modify in some way the predicate that does constitute the main force of the utterance. In terms of subcategorisation requirements and simplicity of derivation, the approach that I have been outlining here offers considerable advantages. The basic lexical entries combined with the <LINK> operation which covers a wide range of data allow the derivation of structure from which particular interpretive effects are achieved. This obviates the need to associate this interpretation with a particular functional projection or to postulate different grammatical roles for classes of lexical items. I discuss how an 'adverbial flavour' may be associated with this derivation, but argue that this depends on the individual predicates. The discussion below concerns adjectives and participles; in 5.4. I discuss noun phrases, which constitute a distinct phenomenon. 5.5. addresses prepositional

phrases, whether or not these may build restrictive L-trees, and the particular problems they raise.

5.3.1. The Adverbial Interpretation

What is meant by saying that the adjunct predicate has an adverbial interpretation? Consider example (96) below. Intuitively it would be natural to say that (96) implies that John sat down and that John was unsatisfied. However, the meaning of (96) goes beyond this.

(96) John sat down unsatisfied.

The adverbial interpretation means that these predicates in some way describe the action which is the main point of the utterance. Intuitively they add some further specification to that action, modifying it in some way. In (96), 'unsatisfied' seems to be describing something about the *action* of sitting down. This is systematically the case in the examples below.

(97) the girls ate the buffet standing.
(98) that man went home sick.
(99) many people watched the film happy.

All of the above can serve as felicitous answers to the relevant question with 'how'.

(100) how did John sit down?
(101) how did the girls eat the buffet?
(102) how did that man go home?
(103) how did many people watch the film?

In the previous chapter, I introduced the idea of assertative force, discussed in Ziv and Cole (1975) and developed from the ideas of Communicative Dynamism of Firbas (1957, 1992).[22] When there are two trees of type (t) in an

[22] I am not adopting this as a theoretical construct. See the idea of main relevance in chapter six. Communicative force has been used to explain various facets of grammatical behaviour. See, for example, Erteschik-Shir (1973), Erteschik-Shir and Lappin (1979).

utterance, then *ceteris paribus* either can carry the assertative force (or indeed both can). This is not the case with these adjunct predicates: they have to play some minor role in the utterance. This is readily demonstrated by means of tag questions.

(104) a. John sat down unsatisfied, didn't he?
b. *John sat down unsatisfied, wasn't he?
(105) a. the girls ate the buffet standing, didn't they?
b. *the girls ate the buffet standing, weren't they?
(106) a. that man went home sick, didn't he?
b. that man went home sick, wasn't he?
(107) a. many people watched the film happy, didn't they?.
b. many people watched the film happy, weren't they?

These results are unsurprising given that these are secondary predications. They show that the 'status' of a predication is intimately linked to tense. (cf the discussion of extraposed relatives in the previous chapter). However, there is nothing intrinsic about secondary predication which will cause an adverbial interpretation.

One way to capture this is to associate the predicate with a certain projection or structural position where the interpretation is linked to a particular function, along the lines of Cinque's (1997) approach to adverbs, where functional heads are claimed to exist for different types of adverbial interpretation. However, in the cases being discussed currently there is no adverbial morphology, so such an approach could only be stipulatory. Moreover, this is not an option for the overall account proposed, where there is no way of representing or motivating such a structural configuration. In the <LINK> analysis the adverbial interpretation has to be derivable from more general factors. There is nothing about the L-tree being non-restrictive that would force this, just because its function is not as a restriction on the interpretation of the noun phrase.

This so-called adverbial reading reflects a certain way of interrelating the clauses. It stems from a degree of connectivity between the clauses, but this is only one of the available options. That there *is* a high degree of connectivity between the clauses is dictated by the structure, but the particular effect that this

has on how the two trees interrelate is determined by their content and by pragmatics. The adverbial modification is only one of the possible ways. I now discuss other interpretations this configuration may lead to.

5.3.2. Beyond the Adverbial Interpretation

On closer investigation it turns out that 'adverbial' is inadequate to describe the interpretive effects that may be achieved with adjunct predicates. I discuss here different cases where the adjunct predicate does not have an adverbial interpretation. This is also confirmed by syntactic tests.

In chapter four I discussed the way in which extraposed relatives could take on the function of introducing a new discourse referent when associated with some so-called predicate of emergence. Above I illustrated how the secondary predicate in those examples could not have the main force of the utterance associated with it. However, it seems to be possible to have examples where the secondary predicate does serve an identificatory function. Compare the examples below. In (108) the function of the secondary predicate seems to be to give additional identifying information about the man who is arriving, while in (109) the secondary predicate is specifying some additional information about the manner of the arriving.

(108) a man will be arriving in the casualty department wounded in a car accident.
(109) a man will be arriving in the casualty department brought here by an air ambulance.

This is confirmed by consideration of how these might be questioned.

(110) how will a man be arriving in the casualty department?

Here (109) provides a felicitous answer but (108) does not.

The basic characteristic is that the secondary predicate holds at the same time as the first. Any interpretive effects beyond this have to stem from the predicate meaning, as the examples in (111)-(114) illustrate. The same sort of truth

conditions apply to all of them, suggesting that there should not be any difference in the logical structure derived for each, but each is subtly different in terms of the ways in which the predicates relate. In (111), it may be inferred that John's exhaustion was caused by the rigours of his journey. The secondary predicate is a consequence of the first. In (112), it may be inferred that the reason for his collapsing is his exhaustion, so the secondary predicate may be said to cause the primary. In (113) there is not necessarily any relation between the two events; it is simply the case that both these properties hold simultaneously to describe John at a particular moment in time. (114) gives an example where the interpretation is more 'purely' adverbial.

(111) John arrived at his destination exhausted.
(112) John collapsed exhausted.
(113) John sat at the back of the room exhausted.
(114) John passed the finishing line exhausted.

(115) - (118) give further examples of these types of reading.

Consequential

(115) John got home from the pub totally drunk.

Causal

(116) John stormed out of the meeting furious at the decision.

Simultaneous

(117) John switched on the radio surprised by the winter sunshine.

Adverbial

(118) John tackled the problem weary.

This is a function of the interpretation assigned in context rather than any intrinsic lexical property. In (119), John's drunkenness may be the reason for his fumbling. However it may be that John has poor motor co-ordination so that he always fumbles for his keys, and that his being drunk may just be another factor. In which case it may be that this is simply an additional property of him at the time, or it may be more of an adverbial.

(119) John fumbled for his keys drunk.
(120) Danny came home from the pub drunk.

In (120) 'drunk' may describe Danny's demeanour as he came home from the pub. He may have decided to leave the pub on account of his drunkenness, or it may be that this is just the state he got home in as is habitually the case.

What these illustrate is that pragmatic factors are at play in deriving the interpretation and that this has to happen within what has traditionally been conceived of as the sentence level. In the analysis I have adopted, this makes sense; each tree potentially constitutes an inferential unit, so the question becomes how to explain their interrelation and pragmatic effects on them. At this stage I have simply articulated some interpretive possibilities; in chapter seven I discuss this further in terms of how structure relates to pragmatic processes.

5.3.3. Adverbial versus Adverb

I return to the problem of the actual semantic requirements of these objects. The present discussion is centred on participles and adjectives. In both cases there is a subcategorisation requirement that they have some subject, in whatever way this is instantiated. In terms of the present approach, this is straightforward, and there is a uniform characterisation of predicative uses of adjectives and participles. These have to be predicated of an entity. Any deviation from this would be subject to the following problems: it would i) induce further type ambiguity; ii) ignore the intuitive subcategorisation requirements; and iii) reduce the analysis back to the position of having some undefined 'adverbial' structure.

This is reflected in terms of the truth conditional entailments of utterances with secondary predicates, as has been demonstrated above. This remains the case regardless of the extent to which these are taken to be adverbials.

(121) Helen left the cinema angry.

In (121) the property of being angry is predicated of Helen. This fact remains despite the possibility of adopting an event variable, as proposed in Davidson (1967), and espoused more recently in Discourse Representation Theory, for which see Kamp and Reyle (1993). It is not the case that 'angry' holds of some event of Helen leaving the cinema.

The argument against the wholesale adoption of an event variable was first put by Fodor (1973) in discussing adverbial modification. Taking the example (122), he points out that the formulation in (123) is not only incorrect but nonsensical: 'to the morning star' is not a property of events.

(122) I flew my spaceship to the morning star.
(123) $(\exists x)(\text{flew}(\text{I, My spaceship, } x))\ \&\ (\text{to}(\text{the morning star, } x))$

Similarly for (121), being angry is not a property of the event. This extends to all of the cases outlined above.

Real adverbials differ in this respect.[23] These do actually modify an event. Where modification is done by an actual adverb, it does say something about the action itself.[24] In (124), there is something angry about the manner of the leaving, where this idea of anger may differ from the idea of anger as applied to a person evoked by 'angry'.

(124) John left angrily

The difference in interpretation is further illustrated below in the contrast between (125) and (126).

---

[23] By this I mean lexical items which cannot have any function other than to modify verbs and/or sentences. In English these are characteristically, but not exclusively, marked by '-ly' morphology.

[24] Note that prepositions seem to be able to project a predicate of either an entity (projected by a noun phrase) or of the event. In (i), it may be that the letter itself is for John or it may be that the event as a whole was on his behalf. This, along with their other functions, is discussed below.

(i) I sent the letter for John.

(125) John sat down happily.
(126) John sat down happy.

In (125) the manner of John's sitting down was happy; in (126) the state he was in as he sat down was happy. The difference lies in what is being modified.

I do not intend here to go into a detailed examination of the semantics of adverbs, and how this is derived in LDSNL. Their interpretation too depends on what it is they modify.[25] The point here is that they are genuinely modifiers. That is to say, they do not change the type specification of the object on which they operate. So if the verb is modified, a composite verb will be the result.[26]

As with other cases of modification, adverbs can only involve a single conceptual unit. This explains the contrast between (127) and (128).

(127) the people left the cinema angry at the film.
(128) *the people left the cinema angrily at the film.

This phenomenon was discussed in chapter three, where I drew the distinction between pre-modifying adjectives and post-modifying adjectives. The former involve a structural operation of modification; the latter one of predication. It is only when predication is involved that the requisite structure is established such that additional material can be inserted. In adverbials, whatever is modified remains the same in terms of the type of object it is, just as with (pre)-modifying adjectives which do not change the type of the noun.

The way in which adjunct predicates acquire an 'adverbial' interpretation is distinct from the processing mechanisms of actual adverbs. The interpretive effects stem from some wider process of combining inferential units. The structure derived for adjunct predicates is an L-tree of type (p). This is a minor premise, and in terms of utterance interpretation is predicted to perform some function other than constitute the main point.

---

[25] See Greenbaum (1969) for a comprehensive description of types of adverbial modification; see McConnell-Ginet (1982) for the syntactic issues involved in verb phrase versus sentence modification; also Gawron (1985) who discusses co-predicating prepositional phrases related to the content of the verb.

[26] So an object of type (e→t) will remain of type (e→t) after modification; similarly an object of type (t) will remain of type (t), and, say, an event variable of type (e) would remain unchanged as type (e).

The only way that <LINK> has been defined thus far is in terms of conjunction. What does this mean? The two predicates differ in terms of what they bring to the process: one of the predicates is marked with tense information and the other one is not. The operation does not result in the creation of a composite predicate; I have dismissed such an approach already as this does not allow any sense of the minor predicate affecting the major one. What the <LINK> operation sets up is two trees, each one complete in itself.[27] One of these is tense marked and constitutes the main part of the utterance; the other, formed by the adjunct predicate, is a minor part. Putting these together will result in some kind of inferential effects, because of the structural and processing connectivity.

5.4. Noun Phrases

Noun phrases can be used as adjunct predicates, but only in specific cases. I examine what these are, and how this relates to the <LINK> operation.

Rapoport (1991) claims that only stage-level predicates can be used as adjunct predicates, as shown in the contrasts in (129) (her 19).

(129) a. Ayala sold the book used.
*Ayala sold the book interesting.
b. Mixa broke the glass new.
*Mixa broke the glass blue.
c. Shuli ate the berries raw.
*Shuli ate the berries large.

On this basis she explains the claimed ungrammaticality of noun phrases as adjunct predicates, as showin in (130), although they are otherwise acceptable as secondary predicates, as in (131). (Rapoport 1991:168-169).

(130) a. *Noa ate the meat a big piece.
b. *Tal sold the tuxedos rags.
c. *Liat read the book a best seller.

[27] That is to say, well-formed according to the requirements of the processing operation, with all lexical information having been used and with no remaining requirements. This does not guarantee well-formedness of an utterance, which relies on contracting wider relations. So, for example, in English a type (p) tree is only acceptable in certain contexts.

(131) I consider Roni a fool.

I do not dispute Rapoport's generalisation that only when predicates have available a stage-level interpretation can they be used as adjunct predicates; however, this can also be true of nouns, in which case nouns can appear as adjunct predicates. Being stage or individual level is not necessarily an intrinsic property of lexical items. While a noun will typically describe an individual, it may be the case that a noun can denote a property that is true in stages; that it is true at a certain time, but not at other times. When noun phrases can be used to project a stage level interpretation then they may be used as adjunct predicates. Consider the examples (132)-(136).

(132) Bibi started a liar and remained a liar.
(133) Tara arrived a New Labour supporter and left a socialist.
(134) Jack started the day the Home Secretary.
(135) Seth left home a boy and returned a man.
(136) Sarah came back from the war an experienced pilot.

In each case these have to have a stage level interpretation where the noun phrase denotes some property of the object that is implied as being at least potentially temporary. That is to say, they form stage level predicates.

It is also possible to construct examples for prototypically individual-level adjectives, denoting properties such as colours, as long as there is some change of state.

(137) The car set off on the journey white; after three weeks' trek through the jungle it rolled up back at the house brown.

These examples as a whole seem to have a resultative 'flavour' to them, but they differ in their properties from resultatives. Unlike resultatives, the predicate has to apply to the subject, as given in (138) overleaf. In resultatives the predicate has to apply to the object, as in (139), and therefore resultatives may require the insertion of some anaphoric element, shown in (140) .

(138) **Terry** left the interview **a wreck**.
(139) Tony drank **the bottle dry**.
(140) John slept **himself sober**.

With a resultative the result predicate does not have to be simultaneous with the main action; rather, as it comes about as the result of that action, it is temporally located at the end of that. The object changes as a result of some action performed on it; that action is described by the verb.

In the case of adjunct predicates, there is still the requirement that the two predicates are true at the same time. Whatever has caused the 'result' described by the adjunct predicate is not described by the initial verb but may be linguistically underspecified and require a large amount of contextual inferencing.

Consider the interpretation of (141), repeated from (136) above.

(141) Sarah came back from the war an experienced pilot.

On first glance, it might seem that the description 'an experienced pilot' applies as a result, and therefore in temporal terms comes after the proposition encoded by the first predicate. However, what is actually being said here is that the description of her as coming back from the war and the description of her as being an experienced pilot are claimed to hold at the same time. Her being an experienced pilot may be the result of the war, but it is not the result of her coming home from the war. This effect is also apparent in the reduced example (142).

(142) she came back an experienced pilot.

Again here there is the implication that the fact that she is an experienced pilot is in some sense due to the experiences she had while she was away, but the two predicates hold simultaneously. This is exactly what is predicted by the <LINK> analysis, and parallels the cases of adjectives and participles described above.

This derivation of an inferential link is not apparent in cases which do not have this structure. In the conjoined case there is a greater degree of separation - by marking tense on both, each one is deemed to have individual importance.[28] Any connection between them holds over and above the point that each of them makes individually.

(143) she came back from the war and she was an experienced pilot.

The same holds when the information is packaged in two sentences. In that case both facts are given, but there is not necessarily any degree of connectivity between them.

(144) she came back from the war. She was an experienced pilot.

The <LINK> analysis for noun phrases as adjunct predicates has them deriving L-trees of type (p). This gives the same structural representation for the basic predicates when they form discontinuous input, and the same inferential effects deriving from this. I now consider the final case, prepositional phrases.

5.5. Prepositional Phrases as Adjunct Predicates

As basic predicates, prepositional phrases should exhibit the same properties as the other cases I have discussed in this chapter. What complicates their analysis is that prepositional phrases can combine in different ways with the main predicate of an utterance. Nonetheless, there is a clear set of cases where discontinuous prepositional phrases give rise to L-trees of type (p). Furthermore, I will show that, in general, prepositional phrases do pattern with other p-predicates in that they can only give rise to non-restrictive L-trees.

5.5.1. Prepositional Phrases in LDSNL

From the theory set out so far, the logical way to approach extraposed prepositional phrases is as <LINK> structures. I have to distinguish those cases

[28] For the effects of 'and' co-ordination on pragmatic processing see Carston (1988, 1998).

in which this is the correct structure to derive the interpretation from those cases where the prepositional phrase is performing some other role. I must therefore further clarify how the various uses of prepositional phrases can be modelled in the current framework.

Note how the following examples differ. In (145) 'to Mary' is not predicated of the giving event. Rather it is part of the action denoted by the verb phrase.

(145) Terry gave the card to Mary.
(146) Terry gave the card to Mary in the kitchen
(147) Terry gave the card to the man in the kitchen.

In (146) 'in the kitchen' is predicated of the giving event, serving to locate the action. In (147), on the reduced relative reading, 'in the kitchen' locates the man.

A prepositional phrase can mark an oblique argument internal to the verb. This is exemplified in (148)-(150), where the prepositional phrase is incorporated into the semantic unit the verb establishes.

(148) John gave the book to Mary.
(149) Janice wrote the letter with a fountain pen.
(150) John came by bicycle.

In these cases the preposition is not predicative, but marks an argument to the verb phrase. This approach is detailed in Marten (in prep.). He argues for underspecification of the subcategorisation requirements of the verb phrase, and implements this in LDSNL using the e-star mechanism.

The preposition is like other lexical categories in that it has context dependent lexical entries. These instantiate predicative and non-predicative update functions depending on the tree environment. A preposition can have a modifying function, where it does not change the type specification of the noun which follows it. In the e* environment, the modifying function of the preposition is triggered. It labels the following noun phrase with relational information but does not change the type of this from type (e). The prepositional phrase is then incorporated as an argument. As modifying uses of the

preposition are dependent on interaction with the semantic content of the verb, I do not discuss them further.

The use of context dependent lexical entries allows for the definition of separate update functions according to the particular environment the structure building process has got to. The phonological material directs the process to a certain address; that is to say, the lexical entry is scanned. The lexical entry can specify what operation to perform according to where the pointer is and what the requirements of the process are. Recall that a lexical rule can only apply when there is the correct triggering environment.

A different source of ambiguity arises with locative prepositions such as 'in', where these may locate an event or an entity. This is illustrated in (151).

(151) the man in the kitchen ate a croissant.
(152) Peter ate a croissant in the kitchen.

In (151) this is a restrictive which serves to locate 'the man'. In (152) this locates the event of Peter's croissant eating. This property is exhibited by other prepositions, and it may be impossible to distinguish them in any meaningful way, as (153) shows.

(153) John left the house with Mary.

In these cases there is a structural ambiguity between whether it is the verb phrase or the noun phrase that is being modified. So, in (154) the prepositional phrase can be taken as a non-restrictive L-tree, with 'John' as the shared node, whereas in (155) this is an L-tree with the whole event carried over.

(154) John left the house in a cardigan.
(155) John left the house in a hurry.

How is it ever possible to distinguish what action a preposition triggers? Whatever structure has been derived will determine what options are later available. If a prepositional phrase is not incorporated as an argument internal to

the verb, then it operates predicatively and builds an L-tree. The shared node has to be identified; but this is the general problem of disambiguation. This can only be determined by the conceptual representation derived in context. All operations are constrained by considerations of relevance, so that any interpretation derived has to achieve adequate contextual effects.

5.5.2. Prepositional Phrases and Non-restrictive L-trees

I now turn to those cases where a non-restrictive L-tree is derived from a prepositional phrase. Examples of this are given in (156)-(159).

(156) the man cooked dinner with a smile on his face.
(157) Mandy sat on the Northern Line in a foul mood.
(158) a man walked in to the classroom with a child on his back
(159) some people tasted the wines with their eyes shut.

These can be straightforwardly derived using the operations outlined. In these cases there is the same interpretive effect as found with other p-predicates. I established that in those cases the L-tree could only be non-restrictive; does the same generalisation hold for prepositions?

Certain constructions with prepositional phrases have been analysed as cases of extraposition (Guéron 1980, Baltin 1983, Culicover and Rochemont 1990).[29] Examples of this are given below.

(160) a book came out by Chomsky.
(161) a man arrived with blonde hair.
(162) a boy showed up with attitude.

In these cases the prepositional phrase is analysed as further modifying the noun phrase, rather than 'changing' the predicate. In the general analysis I have outlined how the basic predicate forms an L-tree and has to combine with an entity in the main clause. It may or may not be the case that adverbial effects are achieved in the interpretation. Should it be necessary, then, to derive structure for the above examples as restrictive L-trees? This would be counter to the

[29] See also the discussion and references in chapter four.

generalisation I have made that in the case of p-predicates as discontinuous input, the L-tree derived has to be non-restrictive. This generalisation stemmed from the fact that the shared node has to be highly accessible (cf 5.2.4.).

There are two arguments which suggest that in fact these are not restrictive L-trees. Firstly, when used with universal quantification, the truth conditions displayed by the prepositional examples are those of non-restrictive L-trees. Secondly, there is a general underdeterminacy in the type of interpretive effects, which is exactly what would be predicted in the case of non-restrictive L-trees, and individual examples display this.

In 5.2.3. I demonstrated using the quantifier 'every' that the basic predicates could not give rise to restrictive L-trees. This is also the case with prepositional phrases. Reconsider the examples from above in terms of universal quantification:

(163) every book came out by Chomsky.
(164) every man arrived with blonde hair.
(165) every boy showed up with attitude.

The first thing to observe about these examples is that in none of these cases can a restrictive structure be derived. (163) does not correspond to (166). Compare also (167) and (168).

(166) every book by Chomsky came out.
(167) every book that was by Chomsky came out.
(168) ?every book came out which was by Chomsky.

The prepositional phrase cannot be restrictive. This is what is predicted for a p-predicate. Moreover, in all of these examples, it seems that the prepositional phrase is in some way 'adverbial'. These have the same requirement for some connection between the predicates.

Why should this differ from the existential cases (160)-(162)? These examples all involve predicates of emergence. In those cases where the main predicate is relatively empty, the identificational function can be derived with the existential. These examples are fine when they serve to introduce a new

discourse entity. The point is that there has to be some inferential effect achieved in the widest sense, but not necessarily that any 'relation' has to be contracted between the two propositions. This sort of interpretation is not available with universal quantification, so there has to be some other kind of relation between the two predicates. The sorts of effects that may be achieved with prepositional phrases parallel those that are displayed by other basic predicates.

In (169) there may be the implication that the men being talked about previously did not have blonde hair.

(169) every man arrived with blonde hair.

In previous chapters I discussed the restrictive/non-restrictive difference in terms of the function performed by an L-tree. I drew a distinction between cases where the referent of the noun phrase was determined by the L-tree and cases where it was not. In the latter case, however, the primary function of the L-tree may still be to provide additional identifying information. In the cases with indefinites one of the possibilities for these non-restrictive L-trees is that they have an identificational function.

In this case there is a tension between the informational structure and the actual syntactic form, in that the main predicate seems to take some secondary form.

(170) every book arrived by Chomsky.
(171) every book arrived at the shop in pristine condition by courier.

This is distinctly odd. The problem here is exactly that the prepositional phrase cannot be construed as restrictive; however, if it is non-restrictive then there ought to be some plausible connection with the main predicate.

Returning to the question of truth conditions illustrates the advantages of the <LINK> approach. Consider the examples below.

(172) every man arrived with blonde hair.

(173) a man arrived with blonde hair.

In the above example, while it may be the case that it has to be true that all the men arriving had blonde hair, this is not a restrictive predicate. The truth conditions are such that the men in question arrive and that they have blonde hair. This is derived from the <LINK> structure. However, it may be that the significance of the 'blonde hair' information is associated with the predicate or with the noun. This is the difference in transformational syntax between having the prepositional phrase as an adjunct to the verb, or as an adjunct to the noun which is then suitably extraposed. It is unclear at what level this difference will have any effect. In the LDSNL analysis, the structure derived is the same in each case, straightforwardly fulfilling the subcategorisation requirements of the prepositional phrase. The way the information contained in the L-tree and that in the main tree relate to each other depends on pragmatic factors.

Prepositional phrases are just like the other p-predicates in that a number of interpretations may appear according to the conceptual content of particular examples.

5.6. Conclusion

The strength of the current approach is that all these structures can be derived by the application of a basic set of rules, all other combinatorial restrictions being derived solely from the information which the lexical item contributes to the process anyway. This results in a much simpler model of 'grammar', where all and only the presented information is relevant for structure. More significantly, the structures thus derived will provide input to a general reasoning domain which in turn is pertinent for the derivation of the interpretive effects displayed by these structures.

I have defined the terms 'restrictive' and 'non-restrictive' in terms of <LINK> structures. This gives a characterisation different from how these terms are generally understood, and allows a more finely grained set of distinctions. The L-trees discussed in this chapter are non-restrictive, in that they are not part

of the e-task. Nonetheless, they are located within the processing domain of the main tree of the utterance, and hence may be bound by quantification, and they have to be integrated with the main tree as secondary predicates of some form.

I now turn my attention to what is more conventionally understood as non-restrictive, the parentheticals. These are also amenable to analysis as <LINK> structures, differing in predictable ways according to the manner in which they are processed.

Chapter Six

Non-Integrated Constituency and <LINK>: Parentheticals

6.0 Introduction

I have discussed how the <LINK> analysis gives a different perspective on the restrictive/non-restrictive distinction; I define this as a property of L-trees which is characterised in terms of the node that is shared between the L-tree and the main tree, and where the <LINK> operation takes place. In chapter five I demonstrated that as discontinuous input, basic predicates form L-trees which are non-restrictive but which fall within the processing domain of the main tree. In this chapter I examine those cases where p-predicates can be analysed as giving rise to L-trees which are not only non-restrictive but which are located outside the processing domain of the main tree. These are the so-called 'parentheticals'.

The problems raised for interpretation by parentheticals are at the crux of the interaction of syntax, semantics and pragmatics. They do not appear to contract any syntactic relations with the main sentence. However, they are dependent on part of that for their interpretation, and so have been characterised as connected more loosely, in some pragmatic way. From the point of view of traditional approaches to syntax they are an anomaly which requires special treatment. For the present framework, they are entirely unsurprising; the general approach that I am proposing is that the structural operation, whatever that may be for a particular (set of) lexical item(s), is the same in both parenthetical and non-parenthetical cases. What differs is where this is located in terms of processing tasks.

I am proposing here that there is a particular way that structure is assigned and that this is the same across all these cases involving 'secondary predication' and the <LINK> operation defined for this. Parenthetical realisations of p-

predicates are a particular instantiation of the <LINK> operation and their existence is predicted as a logical extension of the <LINK> analysis. Their structure and interpretation derives from general principles. Elements are related in the same way, though there are differences in terms of processing and the overall structure of the utterance. I am essentially concerned with predication structures, and how predicates relate to each other in an utterance, rather than operations of modification, where the type specification remains unchanged.[1]

The difficulty is that much of the literature deals with the so-called non-restrictive relative clauses, rather than the p-predicates used parenthetically.[2] However, in the terms of the present discussion these are both parenthetical uses of <LINK> structures. The general issues raised by non-restrictives in terms of levels of structure and the operation of syntax apply equally to other cases where constituents are used parenthetically.

My contention is that the present framework, explicitly adopting the idea that pragmatics has a role to play at the level of the 'grammar', should be able to provide a more integrated account of parentheticals. The <LINK> operation has allowed the derivation of well-formed structures for p-predicates; what challenges does the parenthetical use of p-predicates raise for the <LINK> analysis? Two areas are problematic. The first is the assignment of some kind of temporal location to the proposition in the L-tree and how this relates to the main tree. The second is the flexibility of parentheticals in terms of where they can be positioned in an utterance, and how this can be captured in terms of tree update function. This leads me to reconsider the way in which the <LINK> operation is configured. I propose to split the <LINK> operation into the launching of a new tree task and establishing a relation between two trees, thereby allowing a more detailed examination of the process across different instantiations.

---

[1] As I do not supply a detailed account of how structure building proceeds for *modification* at the sentence level, I accordingly omit an account of parenthetical modification, though I do discuss such cases briefly.

[2] Note that I use the term 'parenthetical' for what others generally refer to as non-restrictive relative clauses. The restrictive/non-restrictive distinction I use solely to describe L-trees, as discussed extensively in previous chapters.

6.1. Characterising Parentheticals

6.1.1. Defining Parentheticals

Relative clauses of the sort given in (1) are the most familiar example of parentheticals. My primary concern in this chapter, however, is the parenthetical use of the basic predicates. Examples of these are given in (2)-(5). I give a technical definition of 'parenthetical' in terms of utterance structure in due course. The following characterisation will suffice as a rough definition for present purposes. Parentheticals are intonationally separated from the rest of the sentence, they are dependent in some way on a part of the main clause for their interpretation, but they are entirely optional units: the main clause is a well-formed sentence without them. (6)-(9) give further examples of what can be a well-formed parenthetical. Note that the properties of having to contract some relation with the main clause for interpretation and being entirely optional are properties which parentheticals share with the cases of <LINK> structures already discussed.

(1) Alex, who Janice likes, watched a film.
(2) Derek, smiling, sat down.
(3) tired out, the children plodded home.
(4) John left, the idiot.
(5) John, from the kitchen, shouted his assent.
(6) Sue, I suppose, reads the Guardian.
(7) he is, according to Jacob, a bit of an adonis.
(8) evidently, Deborah stole the cheese.
(9) Ann talked, of course, about Michael.

None of these examples form a well-formed utterance in isolation.

(10) *who Janice likes.
(11) *smiling.
(12) *tired out.
(13) *the idiot.
(14) *from the kitchen.
(15) *I suppose
(16) *according to Jacob
(17) *evidently
(18) *of course

In (10) the wh- relativiser has to identify with some already established entity; in (11), (12) and (13) the problem is that they are lacking anything to predicate of; that is, there is a basic subcategorisation requirement for a subject which is not met. (14) is just a noun phrase in isolation, and therefore lacks any linguistic function (though unlike the other examples it at least forms a complete object).[3] (15) requires a complement. (16)-(18) are sentence modifiers.

However, the sentences with which they are associated are fine in isolation; that is, they do not in any sense subcategorise for, or require, the parenthetical element.

(19) Alex watched a film.
(20) Derek sat down.
(21) the children plodded home.
(22) John left.
(23) John shouted his assent.
(24) Sue reads the Guardian.
(25) he is a bit of an adonis.
(26) Deborah stole the cheese.
(27) Ann talked about Michael.

I am concerned here with parenthetical constituents that *do* exhibit some form of dependency as condition of their own well-formedness. The examples above differ in this regard from (28) and (29).

(28) Tufnell Park - there was another shooting there last night - seems to be getting more dangerous.
(29) my friend - she's just got a new job - had a party last night.

These are well-formed sentences in their own right, and do not rely essentially on the other sentence for their interpretation. Any anaphoric dependencies there may be are subject to regular discourse constraints, but do not have to be

[3] These can of course be used elliptically, for example in reply to a wh- question as in (i).

(i) Question: What sort of state did Martin turn up in?
Answer: Tired out.

(13), like other noun phrases, can be used if the intended reference and function are sufficiently salient or easy to identify.

guaranteed by syntactic relations.[4] The examples in (28) and (29) are cases of interpolation, where one sentence is inserted into another. What I am concerned with are those cases where there is a requirement for some connection to be derived between the parenthetical and the main clause in order that the parenthetical can be interpreted and have its linguistic requirements fulfilled.[5]

Above I gave examples of p-predicates as parentheticals. The phenomenon of parenthesis extends across phrasal categories, and the subcategorisation requirements of the parenthetical differ of course according to what type of lexical item it is. In the case of the basic predicates, they straightforwardly require a subject. These are like other cases where the basic predicates are used as an additional predicate. As such, they should be amenable to a <LINK> analysis. Examples are given below.

Adjective

(30) Jo, interested by the possibilities, decided to do a bit of research.

Present Participle

(31) the dog, wagging its tail, lapped up the milk.

Past Participle

(32) Martin, stood by the bar, surveyed the crowd.

Nominal

(33) Angela, a strict vegetarian, wouldn't eat a bite.

Prepositional Phrase

(34) Jack, with a big grin on his grin on his face, turned up the music.

---

[4] It may be ultimately that these have to be related by the same sorts of mechanisms as are used to derive linguistic structure, but that is a matter for future research.

[5] Burton-Roberts (1997), as part of a general reappraisal of the nature of linguistic representations, notes that given generally held assumptions in transformational grammar, (i) cannot be described as syntactically ill-formed, as the relation between a non-restrictive relative and its antecedent is not a grammatical one.

(i) *who loves coffee

In the present analysis, however, structure building for *all* these sorts of parentheticals is characterised as a linguistic process.

In the cases of (35) and (36), the modifier *may* be parenthetical, in the sense of providing an extraneous additional constituent.

(35) Matrix, with his customary flair, prepared the latte.
(36) gleefully, the children opened their presents.

However, they do not have to be; it may be the case that they are integrated into the main sentence. This is related to the problems discussed in chapter five about the different structural possibilities associated with prepositional phrases. These may just be cases where the modifier is emphasised by displacement.[6] The crucial feature of <LINK> structures is that they are distinct from the other tree and do not alter its structure as such; it is not apparent that this has to be so for the above. Rather it may be that they perform some update function on the predicate in the main tree.

(37)-(40) give examples of parenthesis which are different again. These involve modification of the whole proposition. The ways in which these sorts of examples contribute to the process of utterance interpretation in Relevance theoretic terms are discussed in detail in Ifantidou (1994).

(37) Sue, I suppose, reads the Guardian.
(38) he is, according to Jacob, a bit of an adonis.
(39) evidently, Deborah stole the cheese.
(40) Ann talked, of course, about Michael.

In these cases, Ifantidou describes the parenthetical element as fine-tuning the illocutionary force of the main utterance.[7] The parenthetical constituent forms an explicature, which does not necessarily form part of the proposition expressed. Rouchota (1996) discusses parentheticals which are taken not to contribute conceptual material, but which she analyses as contributing procedural constraints to the process of interpretation.

---

[6] This is analysed in the present framework as assigning an underspecified tree node location; the underspecification has to be resolved in the subsequent development of the tree. See Kempson and Meyer-Viol (1998).

[7] Hand (1993) claims that in true parentheticals the illocutionary force of the two parts has to be separate.

How should the structure building process be characterised for these? What the <LINK> operation requires is that there be some node which is shared between two trees. This is also the case in these examples. The difference is that in these cases it is the root node of the main clause which is shared. The main focus of the chapter is on cases like (30)-(34), the p-predicates.

6.1.2. The Properties of Parentheticals

Non-restrictive relatives provide the example of parenthetical elements most discussed in the literature; they apparently fail to form syntactic constituents with the main clause, and have a direct, non-parenthetical equivalent, the restrictive relative (see below). Parenthesis is, however, a more general phenomenon, and there are properties common to all examples of parentheticals. Espinal (1991) gives a detailed account of parenthetical phrases, which she refers to as disjunct constituents.[8]

Espinal identifies a number of properties which appear to indicate that the parenthetical does not stand in a syntactic relation to the main sentence (see also Greenbaum (1969) on parenthetical adverbs, McCawley (1982, 1988), Emonds (1979), Safir (1986) on non-restrictive relative clauses). Among the properties she outlines are those outlined below.

Parentheticals do not fill the syntactic position of an argument. In (41) there is no object for the transitive verb 'loves'. In (42) 'Bill' cannot be an argument, (and no obvious parenthetical relation can be derived). In (43), the argument structure of the verb is again saturated, but 'the idiot' can function as a parenthetical.

(41) *John loves, the idiot.
(42) *John loves Mary, Bill.
(43) John loves Mary, the idiot.

The parenthetical and the main clause are independent from each other in terms of what they can contain, so that lexical material may be added to or deleted

[8] Espinal notes the difference between dependent and independent parentheticals, but she does not attach any theoretical significance to this.

from the one without affecting the other. This is shown in (44)-(46), where either the disjunct or the host can be altered, without affecting the other.

(44) John left early last night, the total idiot.
(45) John left early, the idiot.
(46) John left the party early last night, the total utter idiot.

Parentheticals cannot be extracted; they do not form constituents of the main sentence. Therefore, they cannot be focussed in cleft structures.

(47) the children, tired, plodded home.
(48) it was tired that the children plodded home.

Nor can the parenthetical be standardly questioned using wh- questions; that is, it cannot be questioned as part of the sentence.

(49) a. Mary reads, I think, the Guardian.
b.*what do you do that Mary reads the Guardian?
(50) a. Derek, giggling, came in,
b. *what doing did Derek come in?

While (52) is a good answer to (51), (53) is not.

(51) how did the children plod home?
(52) the children plodded home tired out.
(53) the children, tired out, plodded home.

However, it may be possible to question these as parentheticals.

(54) Derek, doing what?, came in.

Espinal argues then that the conceptual objects created by the two clauses are in some way distinct. She presents further evidence that the syntactic structuring of the parts needs to be kept distinct in that sentential operators such as negation do not take scope over the parenthetical element.

(55) Lucy didn't, in the end, show up.
(56) Ralph, satisfied, did not pursue his complaint.

From the current perspective what these scope facts show is that the parenthetical constituent cannot be within the same t-task as the rest of the tree. This is also indicated by the intonational pause. I return to what this means below.

The separation of the parenthetical and the main sentence into distinct units is also apparent in the case of verb phrase anaphora, where it is only the main sentence that is carried over.

(57) Tony, of course, voted Labour, and Paddy did too.

Note, however, that the parenthetical may be carried over - this is a matter of contextual choice. In the example below, it is not clear that it is only the main clause that is carried over.

(58) Tony, thinking of his own best interests, voted Labour, and Paddy did too.

The point remains that there is a reading available which does not include the parenthetical.

Safir (1986) notes that whereas restrictive relative clauses display crossover effects, where a pronoun cannot be co-indexed with the head, non-restrictive relatives are not subject to crossover effects.

(59) The man$_i$ who his$_{*i}$ mother hates [e] arrived yesterday.
(60) The man$_i$, who his$_i$ mother hates [e], arrived yesterday.

Finally, Espinal (op. cit.) notes that parentheticals are intonationally marked off as distinct units using comma intonation.

6.1.3. Parentheticals as a Linguistic Phenomenon

Espinal provides compelling evidence of the fact of separation between the disjunct, or parenthetical constituent, and the rest of the utterance. Nonetheless, whatever sort of analysis is given, the parenthetical element has to combine in some way with (part of) the main sentence. This applies across the different lexical categories, inasmuch as they require some other element in order to be interpretable. In (61) it is Karl who is the teacher. In (62) it is the children who are hungry.

(61) Karl, the teacher, told the class to sit down.
(62) the children, hungry, searched through the fridge.

In the case of sentential modification in (63), it is the proposition that Petra adores Shulamit that is evident.

(63) Petra, evidently, adores Shulamit.

If there is not some kind of connection, then it is not possible to derive any kind of semantic interpretation for these elements. Consider what the truth conditions might be for the above utterances.

(64) (=(the_teacher)(Karl)) & (told.....)(Karl))
(65) (hungry(the_children)) & (searched....(the_children))
(66) (adores(Shulamit)(Petra)) & (evident(adores(Shulamit)(Petra)))

These have to be derivable at some level of semantic representation.

When the basic predicates are used as main clause predicates or secondary predicates, there is no question that they form anything other than linguistic relations with other elements in the sentence. Similarly when adverbs modify the VP as in (67).

(67) John spoke rudely to the man.

The simplest hypothesis, then, for analysing the parenthetical is that the same kind of relation holds as with 'regular' cases of combining the elements, that is to say, deriving structure using the regular linguistic mechanisms by which structure is built and meaning derived. I argue below that exactly this can be achieved by analysing parentheticals in LDSNL using the <LINK> operation. In the account of them that I give, the <LINK> structure involves some element from within the main clause but forms a separate assertion.

Is it necessary for parentheticals to fall within the remit of syntactic theory as data that require explanation? Is it necessary to analyse them as a linguistic phenomenon? These are two separate questions.

There is the general question as to whether or not a linguistic theory should have to account for parentheticals. Could they just be regarded as a performance issue, a matter lying outside the scope of the theory, in the sense that they may be modelled as afterthoughts, corrections or additions? The answer to this is that the parenthetical has to connect to something for its interpretation, or it remains ill-defined and uninterpretable. And in fact what it connects to is something in the main clause. The burden of proof still lies with linguistic theory, in that these phenomena involve linguistic objects interacting with other linguistic objects. Whatever view is taken about how language processing takes place, the fact remains that the ability to assign meaning to parenthetical structures is an integral part of our abilities as language users, and therefore has to be addressed within a theory of linguistic competence.

Nevertheless, as regards the second question, it is not apparent that the phenomenon of parenthesis itself should be characterised as a linguistic one. Firstly, there is no particular reason to believe that parenthesis itself is a well-defined *linguistic* notion in the sense that there is a specific grammatical mechanism to account for it. Secondly, there is the question of whether or not parentheticals can be exhaustively characterised within the syntactic component. For accounts within generative syntax, debate has arisen over whether or not the level at which parentheticals are represented should be characterised as a linguistic one (see below). It has been argued that parentheticals cannot be considered solely within the mechanisms of linguistic processing, and that

although in some sense they form output from the grammar, this has to be mediated with reference to more general cognitive principles.

The present theory offers a different perspective on grammar and cognition, from which the characterisation of parentheticals is not essentially different from general cases of utterance interpretation. The account I present below involves standard linguistic operations, and I claim that these together with more general processing facts are sufficient to account for the data. This avoids altogether the need to have parentheticals at a distinct level of representation. Before I give the details of the LDSNL analysis, I review other approaches to parentheticals and the issues they raise.

6.2. Parentheticals and Syntax

The properties of parentheticals discussed above indicate that they do form units distinct from the rest of the sentence. On the other hand, they still have to be connected to it. A number of solutions to this quandary have been proposed in the literature. Should the parenthetical and the main utterance be connected at a syntactic level, and, if so, what other motivation can be given to justify such a level?

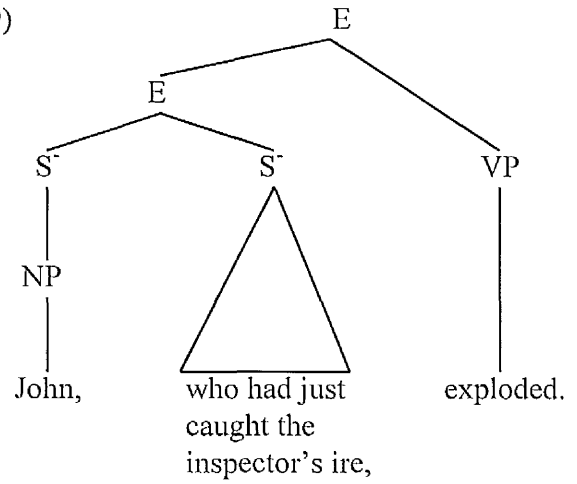
6.2.1. Extra Root Node

In addition to establishing the level at which to attach the parenthetical is the question of where to attach the parenthetical. The problem is compounded by the desirability of avoiding crossing branches where syntactic dependencies cross intervening material. Accounts within transformational grammar have sought to avoid the problem of crossing branches by invoking movement. Some underlying form represents the correct configuration; movement of the constituent gives the correct surface form. Ross (1973) has the parenthetical as a constituent, dominated by the adjacent structure. However, Emonds (1976) claims that the parenthetical is immediately dominated by a super S node at D-structure. This distinguishes the parenthetical as separate from the rest of the utterance. In order to derive the correct surface structure, any necessary material

is extracted from the end of a constituent and moved at S-structure. This avoids the potential problem of having crossing branches, but is stipulatory. Emonds (1979), discussing non-restrictive relative clauses, refines this account. He proposes (69) as the surface structure for (68). Note that in (69) the node marked 'E' represents the initial symbol of the base, which cannot be subordinated, so in some sense can be characterised as a full clause. S-bar (S^{-}) is an intermediate constituent of this parallel, to other cases of X-bar.

(68) John, who had just caught the inspector's ire, exploded.

(69)



This fails to explain why the subject can be divided from its verb by the parenthetical and why normal syntactic processes fail to apply to such an integrated structure.

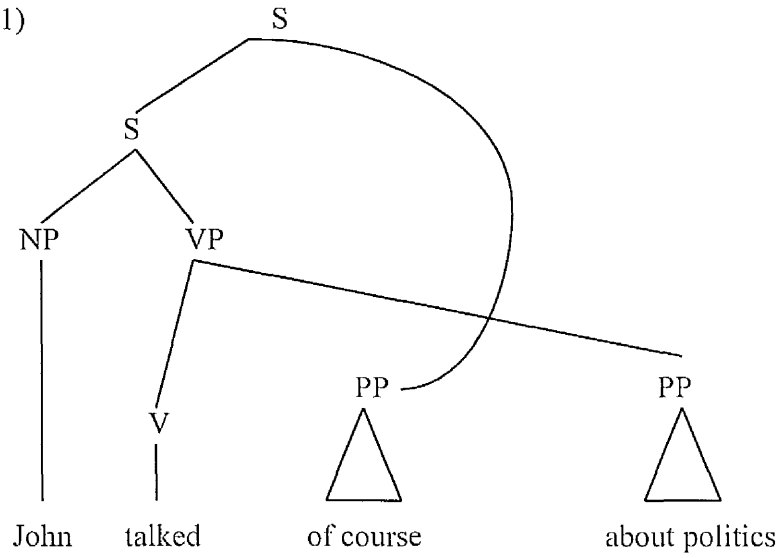
McCawley (1982) is not concerned with the problem of avoiding crossing branches.[9] For example (70) , he proposes the structure in (71), which does not have the parenthetical integrated into the main sentence.

(70) he talked, of course, about politics.

[9] McCawley (1987) provides further arguments for the necessity of allowing discontinuous structure, based on right node raising examples such as (i).

(i) I know that Mary said, and I happen to agree, that she needs a new car.

(71)



The linear ordering determines PF, but at LF, interpretation occurs in the structural position. The non-constituent behaviour noted above is then explained in terms of the parenthetical not being attached within the main hierarchical structure. The problem for this approach is how to map onto an appropriate interpretable structure. What exactly does it mean for the parenthetical to be interpreted at LF in this structural position? Even in the case of the sentential adverbial 'of course' it is not clear what this configuration means in terms of interpretation. In the case of the basic predicates which are my major concern in this chapter, no connection is contracted between the predicate and the argument it needs to take to give the correct interpretation.

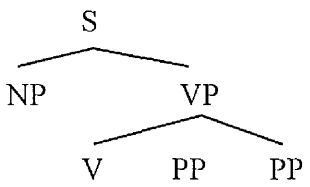
6.2.2. Extra Level of Representation

Safir (1986) argues for a distinction between LF and another linguistic level of LF', at which both non-restrictive relative clauses, and parentheticals in general, are associated with the main structure. At LF these elements are not attached; parentheticals are orphan constituents.[10] Predominantly concerned with relative clauses, Safir claims that restrictive relatives are coindexed at LF, where the Parallelism Constraint on Operator Binding (PCOB) operates producing weak crossover effects. Non-restrictives are not co-indexed until LF', where the

[10] This approach is taken up in Haegemann (1988), who maintains that there is no syntactic relation between the parenthetical and the main sentence.

constraints inducing crossover do not apply. Likewise, VP deletion rules (and presumably negation) are sensitive to the LF representation. He suggests, then, that parentheticals generally can be treated as orphan constituents. The example above would have the structure in (72), ie there is no syntactic attachment at LF.

(72) LF



Orphan constituents are attached at the level of LF' by Attach (α). Safir claims that LF' is a syntactic level. The Principle of Full Interpretation (Chomsky 1986) requires syntactic relations to hold between all elements. The scope facts in (73) provide further confirmation.

(73) John believes that Bill, in his strange way, loves Mary and that Harry does too.

'His' must take 'Bill' as antecedent, and 'in his strange way' cannot apply to John's believing. If 'in his strange way' is not attached structurally, then there should be no scopal restrictions, yet if it is attached at LF, then it would be expected to be part of the antecedent for the VP ellipsis. This is not the case. However, Safir provides no motivation for LF' apart from its being the level at which parenthetical constituents attach.

If LF' is a syntactic level, then what syntactic constraints operate there to restrict the attachment process? Safir states that only adjacent constituents can be the antecedent (for a non-restrictive relative). This is a claim also made by Fabb (1990), based, presumably, in the observation that non-restrictive relative clauses must follow restrictive relative clauses, so that in (74) the whole of the noun phrase is taken as the antecedent. However, it is not correct that the non-restrictive has to be adjacent to its antecedent. I discuss in chapter four how relative clauses may form discontinuous input. The wh- relativiser sets up an

anaphoric choice; this is illustrated in (75)-(80), where the antecedents are marked in bold.

(74) Sally knows the man who arrived from Chorley, who is sitting over there.
(75) I ate all **the biscuits in the cupboard**, which Keith baked yesterday.
(76) **Neil** arrived at lunch time, who I was telling you about this morning.
(77) **Jane** ordered in fluent Chinese, who I had not previously heard speak anything other than English.
(78) he lives in the house on **the hill** by the river, which is called Dumyat.
(79) he lives in **the house** on the hill by the river, which is called Dumyat.
(80) they looked round **a house**, on the hill by the river, the architecture of which would be described as neo-Victorian.

Moreover, what does 'adjacent' mean in the sort of analysis where attachment only occurs at LF'; at what level of representation should the adjacency requirement hold? A simple intuitive idea of linear adjacency would rule out even (75), as in that case 'the cupboard' would have to be the antecedent. However, it cannot be adjacency at surface structure or PF, as no syntactic relation should be contracted there between the parenthetical and the main sentence. Similarly with LF, no syntactic relations are contracted there between the orphan constituent and the rest of the sentence. If the pertinent level is LF', then the restriction becomes entirely circular if whatever happens to be the antecedent is to be characterised as adjacent. That is to say that prior to attachment, there can be no adjacency, as there is no syntactic relation, so adjacency can only be defined at LF' after attachment.

Other parentheticals are still more problematic with regard to adjacency. Here the requirement for adjacency does not apply at all.

(81) smiling, Betty explained the situation.
(82) the children slumped in front of the television, tired out from playing in the garden, and watched Barney.

It is unclear what licenses the formation of orphan constituents at other levels of the syntax (eg LF), where presumably they require representation regardless of

where they contract relations with the main sentence. Nor does there appear to be any other motivation for LF'.

Fabb (1990), who is concerned with the differences between restrictive and non-restrictive relative clauses, adopts a similar approach to Safir (op. cit.). He reformulates LF' as X-structure, but claims that this is a *post*-syntactic level. Fabb distinguishes between the predication relation found in restrictive relatives, which can only take a projection of N', and an 'aboutness' relation in the case of non-restrictives which is not part of the syntax. This explains, he says, why non-restrictives can take any phrasal projection as antecedent. X structure is a discourse level where sentences combine - it is here that the relative pronoun picks up its antecedent through co-indexing. His account is open to the same criticisms as that of Safir. He does not explain how a non-restrictive relative can initially be licensed as part of a syntactic representation, why co-indexing, an erstwhile syntactic operation should happen at a discourse level, nor does it analyse why there should be a need for LF and X-structure.

The need to claim the existence of different levels is motivated purely by theory internal considerations. However, these types of multi-level analysis are problematic given more recent developments in the transformational-generative paradigm which posit PF and LF as the only two levels of syntactic representation (cf Chomsky (1995) and work inspired by that). Moreover, as Burton-Roberts (1997) discusses at length, within the assumptions adopted in transformational grammar there remains a constant tension as to the way that parentheticals should actually be related as *syntactic* constituents.[11]

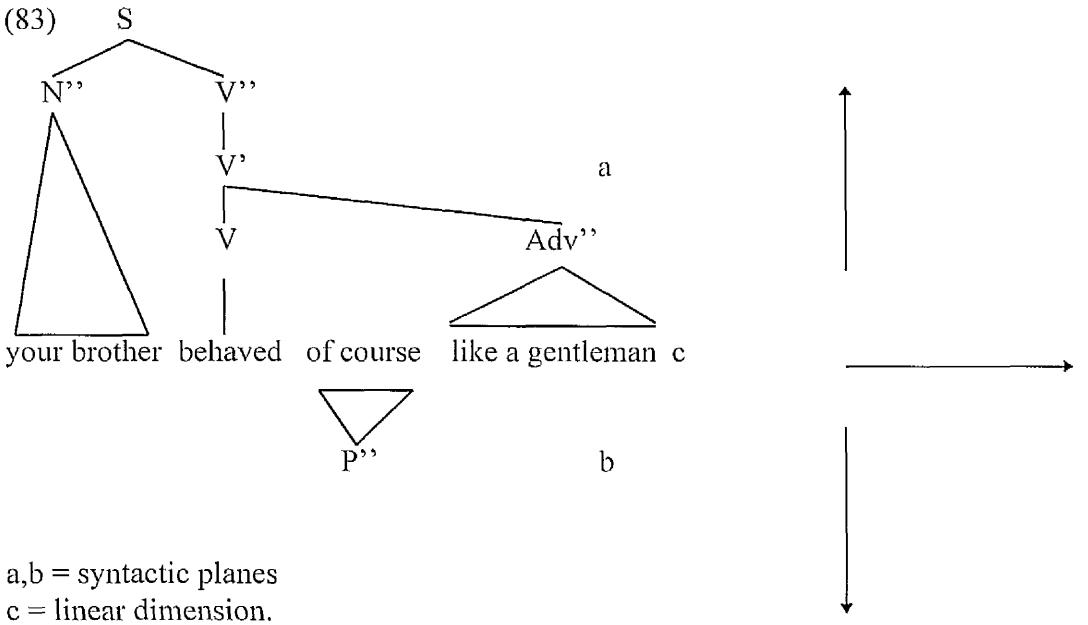
6.2.3. Syntactic Non-Integration

The most radical approach is taken by Espinal (1991), who attempts to dispense with syntactic integration. She deals with the problem of parentheticals in general, rather than restricting attention to a particular subset of them, though she adopts the term 'disjunct constituent'. To account for these, she proposes the

[11] His own proposals involve distinguishing linear precedence constraints (which are not linguistic) from hierarchical dominance relations. This is articulated within a representational theory which reconfigures the approach to syntax by sharply contrasting linguistic expressions from representations of such expressions. See Burton-Roberts (1994) for an introduction to this.

idea of a three dimensional syntactic representation. Parentheticals do not enter into hierarchical relations with their host, rather independent phrase markers may be projected onto separate planes. These intersect on a linear axis which determines surface order, as in (83).

(83)



a,b = syntactic planes
c = linear dimension.

No syntactic relations hold between phrase markers on different planes, and they do not have to be syntactically licensed with regard to each other at the output of the grammar. Rather, each phrase marker is mapped onto a conceptual representation and relations between these are licensed in the language of thought without constraint by linguistic information, but subject to mapping and saturation conditions, as given in (84).

(84)

"Condition on Mapping
Each major syntactic constituent of a multi-rooted LF maps into a conceptual constituent in the conceptual representation corresponding to that LF.
Condition on Saturation
A conceptual structure (including any one corresponding to a complex LF allocated in a three dimensional space) must be saturated, that is, all of its argument places must be filled"

These replace a syntactic principle of full interpretation (Chomsky 1986), as this latter would rule out interpretation of constituents which cannot be licensed at the output of the grammar.[12]

Espinal argues that at the conceptual level established syntactic relations are interpreted (function-argument application and modification); COMMENT relations are established according to conceptual well-formedness requirements and considerations of Relevance (Sperber & Wilson 1986/95) over ontological rather than syntactic categories. For example, an appositive adjectival phrase such as (85) would be licensed as a comment on the conceptual constituent of the host, while a nonrestrictive relative such as (86) would be licensed as a comment on the antecedent of the relative pronoun.

(85) Terry, delighted, accepted the invitation.
(86) Archie, who is a keen swimmer, goes to the pool every day.

I have argued that pragmatics plays a crucial role in determining the way that constituents relate to each other, and in terms of additional aspects of meaning derived. However, Espinal's approach to parentheticals fails to incorporate any notion of linguistic restriction on the way that parentheticals can combine with the host. This is due to the syntactic assumptions she adopts.

The danger of this approach is that it reduces syntax to determining the internal structure of phrases with no particular claim to determining the relations between them. It is possible to talk of well-formed constituents such as NP, VP, PP, but what would the status of these be in isolation, as the sole constituent of a syntactic plane? This is exactly what three dimensional syntax allows.

The linguistic component *could* determine specific modes of combination; indeed on certain planes it still does in Espinal's account. However, the system outlined will allow for anything, as long as it is pragmatically permissible. There is no requirement for the surface form to bear any relation to the actual conceptual structure. Linguistic information cannot, by definition, contribute to the process of combination at the level of conceptual structure. Semantic well-

[12] It seems to me that a by-product of this approach is to undermine the conception of PF as a level of grammar.

formedness ceases to exist as a linguistic requirement. This represents a serious loss of information, in terms of the way in which structure is built. The linguistic information that is used in deriving structure should always be available to the process by which structure is derived, yet the approach that Espinal proposes explicitly rules this out.

The biggest problem with this approach is that the combining of separate syntactic units is blind, by definition, to their syntactic properties; a problem which is not, I believe, mitigated by the notion of conceptual saturation. Why introduce conceptual saturation at all? The idea of conceptual saturation is forced by sticking to a generative characterisation which requires of a complex syntactic structure that the relations between the units of the linear dimension are determined post-linguistically. The notion of syntactic saturation has to be abandoned for certain constituents as the wider conception of syntax will not allow it, where there is no way to integrate syntactic and conceptual well-formedness requirements.

Although differing in their specific solutions, the above accounts illustrate the general problems that have to be faced in accounting for dependent parentheticals. They do not regularly combine in hierarchical relations with their host. Nevertheless, they must contract some relation with their host in order to be interpretable. If this is a linguistic relation, what explains their apparent independence, and what implications does this have for notions of linguistic constraint? Specifically, how does this interact with any putative theta criterion? If this is not a linguistic relation, then how can dependent parentheticals be licensed by the grammar at all? Analyses which position the problem exclusively at a discourse level may be plausible for fully independent sentences, but they raise intractable problems when parentheticals are dependent on the host to resolve their interpretation.

The approach I adopt is not that there is a phenomenon called parenthesis involving a specific grammatical mechanism, but that the same, regular procedures are used. I argue that parentheticals are assigned structure by the linguistic component according to lexically supplied information. Crucially, however, they form separate propositions, and this explains their properties

without having to postulate distinct levels of interpretation. As discussed in previous chapters, structural restrictions and conceptual well-formedness requirements go hand in hand. Defining structure over lexical items rather than as a direct mapping from string onto representation is a defining feature of the framework adopted, LDSNL.

6.3. Structure Building, Parsing Strategies and Parenthetical Constituency

In the above chapters, I demonstrate how a single utterance may result in the building of several trees, where these trees are linked by sharing a node description. This happens in cases where there is predication in the sentence in addition to the main predicate. These secondary predications can be formed by basic predicates which do not supply tense information. I noted in 6.1. that all of these form good parentheticals. I now consider if the same process is taking place whether these additional predications are found within the sentence or as parentheticals. My hypothesis is that all of these cases involve the building of a <LINK> structure, where some node is shared between the trees.

This approach implies that the general properties of the structure-building operation should be the same, and there should be no requirement in this case to define separately some notion of well-formedness for L-trees, given that these already exist as well-formed objects. The relation between L-trees and the main tree has been characterised as determined pragmatically thus far, as a general aspect of the <LINK> operation. However, there are certain idiosyncrasies about the properties of parentheticals which stem from the fact that they exist as separate units for the pragmatic processing of the utterance. This has implications for the way that utterance structure is built by <LINK>, as I discuss below. I start by examining how <LINK> can be applied to cases of type (p) parentheticals.

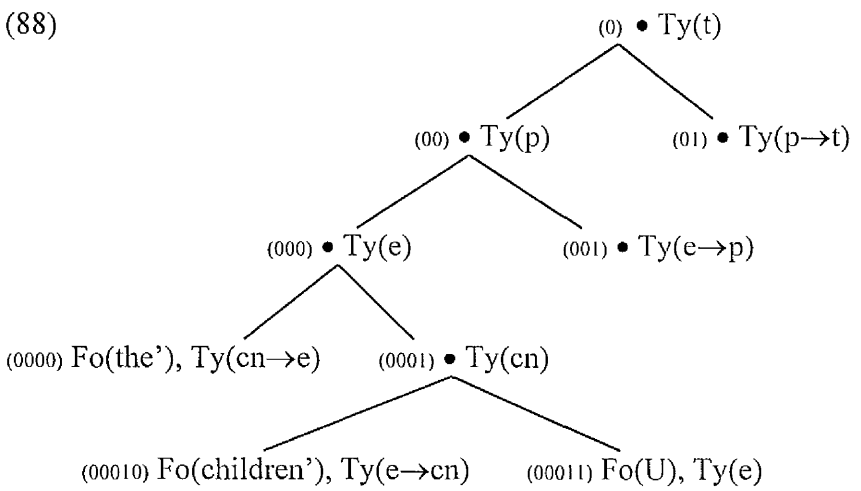
6.3.1. A First Approach

I first set out how structure building might proceed for parentheticals, and what issues this raises for the analysis. I describe the process for the example in (87). To start off with, structure building proceeds as normal.

(87) the children, tired out, plodded home.

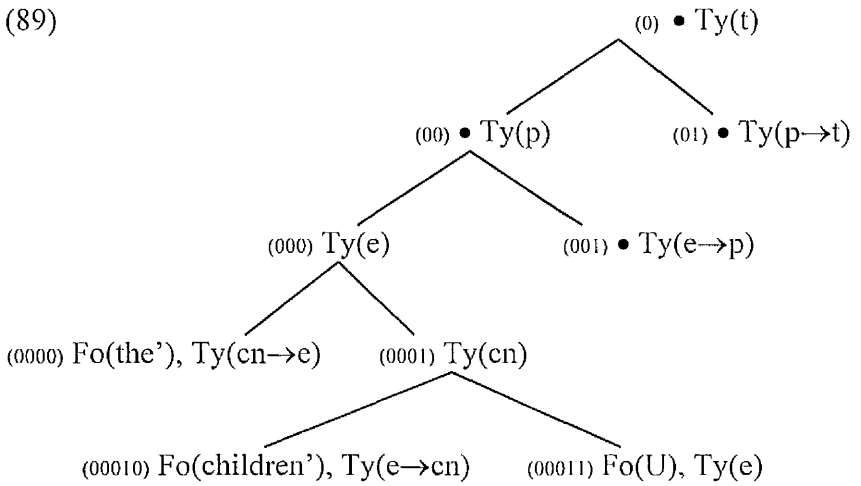
The overall goal is for a truth evaluable proposition, of type (t). The subgoal is for a proposition type (p); this can be broken down into requirements of type (e) and of type (e→p). The initial lexical input, 'the children', is scanned. The structure shown in (88) is built, according to the lexical information supplied by the determiner and the noun.

(88)



At this point in the process there is a pause. This indicates that the current processing task has been completed, and so compilation occurs in the e-node, giving (89).

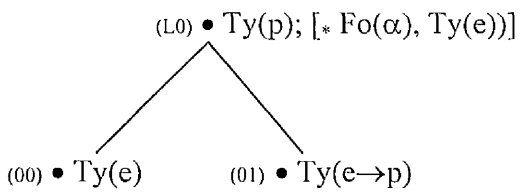
(89)



At this stage the pointer is at (000).

Now the <LINK> operation is launched, which results in a new tree structure as shown in (90), with the requirement to build a tree of type (p).

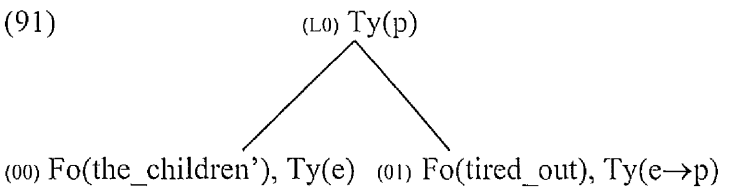
(90)



The <LINK> operation requires the identification of some shared node. In the current case this will be the e-node, where the pointer is in the main tree.

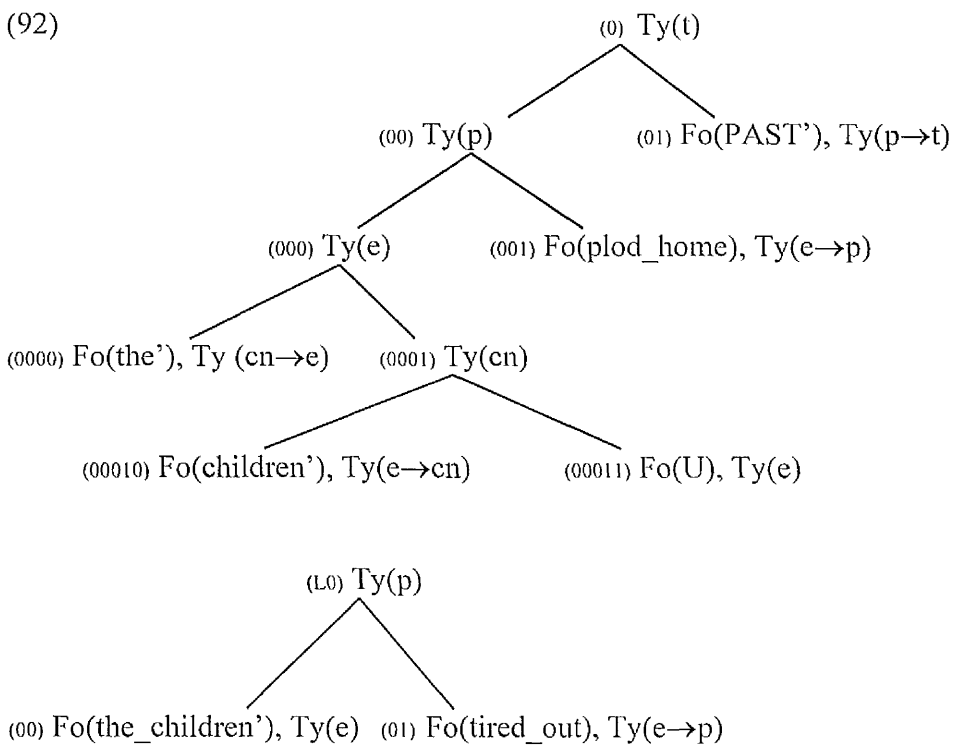
The underlocation of the shared node in the L-tree is resolved as it becomes the subject at location (00). The lexical input scanned is 'tired out'. The information this provides duly annotates the node (01). Compilation takes place and the result is a well-formed L-tree where all the requirements have been fulfilled, as shown in (91).

(91)



The pointer returns to the main tree, at location (000) from where the task of launching the L-tree was initiated. The task of deriving the main tree resumes. The pointer moves down to tree node (001) by the predicate rule (cf 3.6.1.). The next piece of lexical input scanned is the verb 'plodded home'.[13] This annotates node (001), and supplies tense information at node (01). All the lexical information has been used and all the requirements on the tree have been fulfilled. The overall result is (92).

(92)



The above gives the basic way in which application of the <LINK> operation might assign structure to a parenthetical constituent. In this case the derivation of the <LINK> structure occurs outside any processing tasks within the main tree. This is indicated in the input by the break effected by the 'comma intonation', and is reflected in the properties displayed by parentheticals discussed above. What does this mean in the current framework? The parenthetical case is different from those cases of discontinuous input discussed

[13] I omit the internal structure of this as it is not pertinent.

in chapter five, where the <LINK> operation was launched within the p-task. However, this difference is not reflected in the output structure given above.

6.3.2. Processing Tasks and Truth Conditions

What does it mean to talk of operations happening within or outside certain processing tasks? These tasks reflect stages of structure building and the manner of combination of the parts of structure projected as a result of lexical information. The interrelation of the tree nodes reflects the mode of semantic combination, as shown in the truth-conditional differences. In the above derivation I separated off the parenthetical as a distinct processing task; this means that it supplies a proposition independent of the main tree. This should be reflected by some difference in the truth-conditional representation.

In previous chapters I use universal quantification to investigate the ways in which structural dependencies are drawn, and the truth conditions associated with them. This can be explained by the order in which actions are performed as related to task states. The interpretation of the parenthetical differs both from the cases in the previous chapter where the predicate was incorporated into the main processing task, and from the restrictive cases. The possible configurations and the truth conditions they map onto are given below.

Restrictive

(93) every girl disappointed by her exam score entered the classroom.
(94) $\forall(x)((girl(x) \ \& \ disappointed...(x)) \rightarrow entered...(x))$

Non-restrictive

(95) every girl entered the classroom disappointed by her exam score.
(96) $\forall(x)(girl(x) \rightarrow (entered...(x) \ \& \ disappointed(x)))$

Non-integrated/Parenthetical

(97) every girl, disappointed by her exam score, entered the classroom.
(98) $\forall(x)(girl(x) \rightarrow entered...(x)) \ \& \ \forall(x)(girl(x) \rightarrow disappointed(x))$[14]

[14] Equivalent to (i).

(i) $\forall(x)((girl(x) \rightarrow entered(x)) \ \& \ (girl(x) \rightarrow disappointed(x)))$

The result here for the non-integrated example is the establishment of two distinct propositions, each of which is separately evaluable.

(96) and (98) will in fact pick out the same set of objects,[15] but notice that the logical implications and interpretive effects of these may differ. This reflects the difference in terms of the communicative intention. In any particular situation both conditions will hold of all members of the set picked out by the predicate 'girl'; for (95) being disappointed is bound up with entering; (97) need have no such connection; the fact of their being disappointed is presented as a separate assertion in the case of the parenthetical.

In (96), the two predicates are conjoined before the quantificational dependencies are established; for this integrated example, the building of the L-tree is part of the overall processing task; the evaluation of the L-tree is an integral part of evaluating the general meaning; this is where the adverbial reading comes from inasmuch as the two have to be simultaneous

In (98) there are two distinct propositions. In the parenthetical example the L-tree is built as a separate object on-line; it is to be evaluated separately, as a distinct assertion (though part of the same utterance).

For some speakers, quantificational force cannot be carried across into the parenthetical. My own intuitions do not correspond with this, but it is easy to see why this would be the case. On such a reading it is the witness set picked out by the quantificational phrase which the tau term represents in the L-tree. The interpretation of the tau-term in the main tree can only be computed at the t-node once compilation has occurred for the whole tree. That interpretation duly replaces the tau term in the parenthetical.

The evidence above indeed suggests that the process of building <LINK> structures gives rise to distinct assertions; that is to say, non-integrated L-trees which lie outside the t-task of the main tree. If an L-tree lies outside the t-task of the main utterance, how is it to be characterised in terms of type?

[15]The declarative unit derived for the node of type (e) has to be the same in order for the <LINK> relation to hold. Note, though, that it seems that the non-integrated L-tree must be of type (t) in order for quantificational relations to be processed.

6.3.3. Parentheticals and Separation

The parentheticals I am discussing here are all p-predicates. The p-predicates do not supply tense to the proposition that they create. However, the status of the proposition of type (p) is such that it cannot be independently evaluated, as it lacks any temporal location. In the cases of the p-predicates discussed in previous chapters, they relied on the main clause for temporal indexing. But in the current case there is clear evidence that they are not dependent on the main clause in this way. Were they to inherit tense in the same way then they would be predicted to be susceptible to other sentence-level processes. This is not in fact the case with parentheticals, however.

Can it really be the case that the parentheticals constitute a separate t-task? What militates against this view is that in interpretive terms they have to be of type (t), but they do not have to have tense lexically supplied. Nonetheless, the evidence suggests that they do have to be of type (t). The parenthetical can be temporally independent of the main clause. This is unsurprising in the case of parenthetical relative clauses, where tense is supplied by the verb in the clause.

(99) Jack, who drove here, likes champagne.

However, it is equally true for the p-predicate parentheticals.

(100) Terry, a sailor in the second world war, works for the Home Office.

Not only are they distinguishable in terms of tense, they may be completely separate assertions.

(101) Sam said that Jenny, the old fool, has taken Sandra back.

In fact these parentheticals can even be used with different types of speech act.[16]

[16] The parenthetical itself can only be an assertion, however:

(i) * John, acting badly?, performed in Kilburn last night.

(102) did you see John, the old ham, in that play in Kilburn?

These are all properties associated with the t-task.[17] It is these sorts of considerations that led Espinal (op.cit.) to analyse parentheticals as licensed by a COMMENT relation. But this degree of syntactic 'independence' is not always exhibited by parentheticals.

6.3.4. Parentheticals and Integration

The complicating factor is that the parentheticals do not have to be separate in this way. The parenthetical may form a separate assertion,[18] but equally it may not. The truth conditions of the proposition that the parenthetical maps onto can be evaluated separately from the main clause, but the truth conditions of the utterance as a whole depend on both conjuncts being true.

(103) disgusted, John left.
(104) disgusted(john) & left(john)

Annear Thompson (1971), following Bach (1968), argues that it is not always the case that non-restrictive relative clauses[19] always form a separate assertion, as the examples below show (her (44) and (45)). In (105) it may be either the speaker or Harold who thinks his girlfriend crazy; in (106) it may be either the speaker or the claims agent who considers that the paint job should have been done long ago.

(105) Harold says that his girlfriend, who is a little bit crazy, wants to go to Hanoi.
(106) the claims agent said that the paint job, which should have been done long ago, would cost $150.

---

[17] Whatever account is given of these phenomena, they range over propositions of type (t).
[18] Clearly this depends on what is meant by 'assertion'. I am claiming that communicating both propositions may be equally important to the speaker.
[19] That is, parenthetical relatives in the terminology I am using.

The same property holds for the type (p) parentheticals, as illustrated in the examples below. In (107) the assertion that the paint job was carried out sloppily might be being made either by the claims agent or by the speaker.

(107) the claims agent said that the paint job, carried out sloppily by some cowboy, would cost $150.

In fact the preferred reading here seems to me the one where the parenthetical is 'attached' lower down, as part of what the claims agent said. In this case it has to be located somehow within the t-task initiated by 'that'.

(108) and (109) give examples where the tense evaluation is the same.

(108) Jerry, a sailor, lived in Portsmouth.
(109) Mary, delighted, unwrapped the present.

(110) and (111) show parentheticals embedded as part of what is questioned.

(110) have you heard that, fearing the worst, John ran away?
(111) did Sid, furious at the slaughter of his livestock, shoot the fox?

This optionality is predicted by the processing task account: the parenthetical does exist as a separate tree of type (t) and hence may be independently evaluable, but it is equally the case that it may inherit information from the tree to which it is linked. The <LINK> relation indicates solely that there is a node shared between the two trees, and does not indicate that any other kind of structural dependency is necessary.[20] This is seen more clearly in 6.5., where I reconfigure the <LINK> operation. First I have to consider what it means for an L-tree to be of type (t).

6.3.5. Tense and Temporal Location

Any parenthetical can be separately evaluated; structure building should therefore be characterised as a process of introducing a tree with the requirement

[20] Though the connection is associated with interpretive effects according to the way the particular <LINK> structure is established.

of type (t). What this means is that the parenthetical has to have some kind of temporal index; that is to say, it has to be of type (t) but this does not have to be acquired from the main tree through the tense specification of that verb. The parenthetical use of a predicate is defined as when the structure derived is a non-integrated L-tree of type (t).

Thus far I have not discussed the difference between tense and temporal index. A proposition of type (t) has a temporal index; that does not mean that it has to have tense information lexically supplied. By temporal index I mean some label to indicate a location in time. The ways in which a temporal index can be acquired in natural language are varied. (112) shows that the temporal index (for the parenthetical) may be acquired from a prepositional phrase. There are a range of ways in which time location can be specified. Noun phrases can also be used, as (113) demonstrates.

(112) John, a sailor in the last war, is now a minister.
(113) Jo forgot to put the cat out yesterday.

Tense information underspecifies the eventual temporal index derived for an utterance, as (114) and (115) show. These have the same tense information but very different temporal locations.

(114) Anna ate ice-cream five years ago.
(115) Anna ate ice-cream for breakfast.

This is a complex area, the exploration of which is beyond the remit of the present work.[21] All I want to suggest here is that the acquisition of a temporal index by a non-integrated L-tree does not have to rely on lexically supplied tense information. The exact way in which the temporal index is acquired is not obvious, but that is a characteristic of all trees, and so is unsurprising.[22]

Being of type (t) indicates that a proposition is truth evaluable. The type (t) specification indicates that it can provide a vehicle for inference, and that it does

[21] Perrett (in prep.) gives a detailed account of tense in LDSNL.
[22] The non-integrated L-tree may have the same temporal index as the main tree, but this is not acquired by being located within the same processing domain.

not depend on the main utterance to derive its temporal location. In terms of the well-formedness of the utterance as a whole, the structure which the parenthetical maps onto must be licensed in some way, and this happens by dint of it being an L-tree.

In the cases where tense is lexically supplied, I have modelled it as creating a node of type (p→t). It is not obvious that this has to be the case in all instances of acquiring a temporal index, but inasmuch as the temporal index can map from a type (p) proposition to a type (t) proposition I maintain this approach.[23] Note also, it is not the case that there has to be any kind of tense marking on the utterance overall; that is to say, there is no super node of type 't' which dominates both trees. There is no relation between the trees except the <LINK> one.

6.4. The Positioning of Parentheticals

The next issue in terms of developing the <LINK> account for parentheticals concerns where in the linear flow of the sentence they can be positioned. I now consider the implications for the present account of the restrictions on the positioning and interpretation of parentheticals. In the present case I am concerned with the p-predicates. These are more limited in terms of where they can be put than other types of parentheticals which are modifiers or require a type (t) complement. The latter I discuss below. Espinal makes the general claim that parentheticals cannot disrupt phrase structure. (116)-(118) show the positioning possibilities of type (p) parentheticals. They can be placed utterance initially, as in (116), utterance medially, (117), or utterance finally, (118).

(116) tired out by their crying, Reggie slumped in a chair.
(117) Reggie, tired out by their crying, slumped in a chair.
(118) Reggie slumped in a chair, tired out by their crying.

[23] It may be the case in terms of future development of the theory that a more explicit modelling of the way in which higher level explicatures (see Sperber and Wilson 1986/1995) are structured will shed light on tense and temporal indexing. A possible approach would be to handle time/tense in these terms, as distinct from the proposition, though as a compulsory part of that explicature termed the propositional form (ie that over which the truth conditions are defined). What suggests that such an approach might be worth exploring is the way in which parentheticals may apparently differ in terms of speech act/explicature status.

(119) shows the other possibility for medial placement.

(119) ??Reggie slumped, tired out by their crying, in a chair.

(119) seems somewhat odd, breaking up as it does the verb phrase. This effect is displayed more starkly in the case where a parenthetical interrupts the process of a verb combining with a direct object: (120) and (121) are fine, but (122) is not.

(120) sick of his taunts, Graeme hit Robbie.
(121) Graeme, sick of his taunts, hit Robbie.
(122) *Graeme hit, sick of his taunts, Robbie.

This makes sense in that, by launching a new tree, the parenthetical is disrupting the process of structure building for the current one.

The utterance initial parenthetical will prove problematic for the existing <LINK> rule; I discuss this below.

6.4.1. Positioning and Interpretation

What effect does the position of the parenthetical have on the interpretation, ie on the question of what node can be identified as the one shared with the <LINK> tree? Does the L-tree always have to identify with the subject? This is not the case. Consider the data below. I have marked in bold the noun phrase that provides the subject of the parenthetical predicate.

(123) Robbie was hit by **Graeme**, sick of his taunts.
(124) Yugoslavia was attacked by **NATO**, unwilling to prolong the peace talks.
(125) NATO attacked **Yugoslavia**, unwilling to agree at Rambouillet.

When the parenthetical is utterance final it can pick up on any of the preceding noun phrases. When it is medial, it can pick up on any preceding noun phrase, but not on anything that follows. (126)-(127) illustrate this point.

(126) **Graeme** hit Robbie, sick of his taunts, and the referee.

(127) Graeme hit **Robbie**, homophobe and general cretin, and the referee.

In (126) the parenthetical refers to Graeme; in (127) to Robbie, but in neither case can it refer to the referee; so that it is not the case that a parenthetical can identify with any node in the main tree that is created after the <LINK> operation has been launched. As far as these are concerned, the <LINK> rule given in (128) will give the right results.

(128) <LINK> Rule for P-Predication (cf. 5.2.1.)

$$\frac{[_m..[_n\ Fo(\alpha),Ty(e)]..[_o\ Ty(y)]..\bullet\ Ty(t)]}{[_m..[_n\ Fo(\alpha),Ty(e)]..[_o\ Ty(y)]..\bullet\ Ty(t)],[_L\ \bullet\ Ty(p);\ [_*\ Fo(\alpha),\ Ty(e))]\ ]}$$

The rule as stated requires that there be some node already created in the existing tree as the antecedent and precludes the possibility of having anything come later. Moreover, the same restrictions apply in terms of pointer movement, and identifying the shared node. The L-tree which the parenthetical builds has to be non-restrictive.

As already noted, however, parentheticals can occur sentence initially. In this case the parenthetical has to be identified with the subject of the main sentence. In the case of utterance initial parenthetical p-predicates these have to take the subject as the first node.

(129) unwilling to come to agreement, **Yugoslavia** was attacked by Nato.
(130) *unwilling to come to agreement, Nato attacked **Yugoslavia**.
(131) *sick of his taunts, Robbie was hit by **Graeme**.

Moreover, the <LINK> rule as specified cannot operate in the structural context established by a sentence initial parenthetical.

How should this case be characterised in terms of the <LINK> operation? What sort of definition can be given for <LINK> in order to cover these cases? Should this be characterised as some kind of specific construction distinct from the more general cases of <LINK>?

6.4.2. Utterance Initial Parentheticals

How is structure derived when parenthetical predicates are sentence initial? In such cases the regular <LINK> rule cannot apply as the input structural configuration is incorrect; that is, it cannot apply, as there is no other tree already existing and no existing node within such a tree, such that that node can be carried over. In the previous cases of parentheticals I have considered, the node that is shared between the two trees already exists at the point at which the <LINK> operation is launched. The purpose of the <LINK> operation is that it allows the development of a new tree within an existing tree task; ie it allows the specification of a new goal, and steps to be taken to achieve that goal, before the existing goal has been reached. What characterises the configuration with the initial parenthetical, however, is that no significant tree development has yet been achieved. Note that the requirement contained in the <LINK> rule is that some node description should be completed. In the present case, however, the L-tree is anticipatory of the shared node.

6.4.2.1. The Problem of Building Structure

I examine, then, what happens in the case of this parenthetical. How does the structure building process go forward here? The problem is twofold: how to connect the L-tree to the main tree, and how to build the structure in the first place, when it is lacking an element. That is to say, there are not sufficient premises to build a structure when it is the initial element. The subject node of the parenthetical is annotated with a requirement, so it requires some further operation before it can be characterised as well-formed. This clarifies the issue of the connection to the main tree: the parenthetical cannot exist as a well-formed unit until such time as it has inherited a node description from the main tree, and thereby established a <LINK> relation with it.

I consider how structure building proceeds for the example in (131).

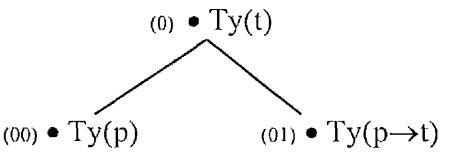
(131) tired out, John slumped into the chair.

The start point for the parenthetical is the same as the general starting point, that is, the requirement for a tree annotated as type (t).

(132) $_{(0)}$ • Ty(t)

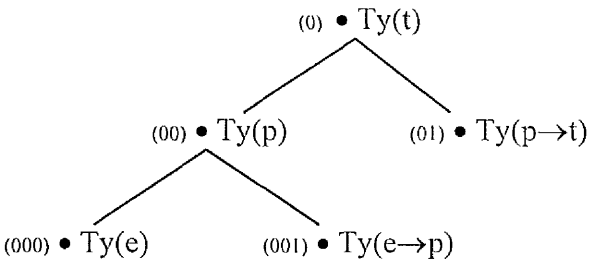
As normal, this can be broken down into the requirements for a predicate and some tense specification.

(133)



The requirement for type (p) at (00) can in turn be broken down into requirements for a subject and a predicate, as shown in (134). These steps are all achieved by the rules of Introduction and Prediction.

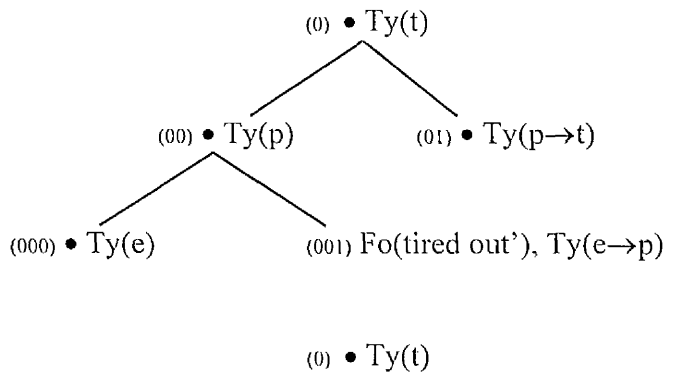
(134)



Lexical input is scanned, and provides a predicate. Moving the pointer to (001) gives the right trigger for the lexical action and the predicate can duly be annotated as a description at (001). The pointer returns to (00).

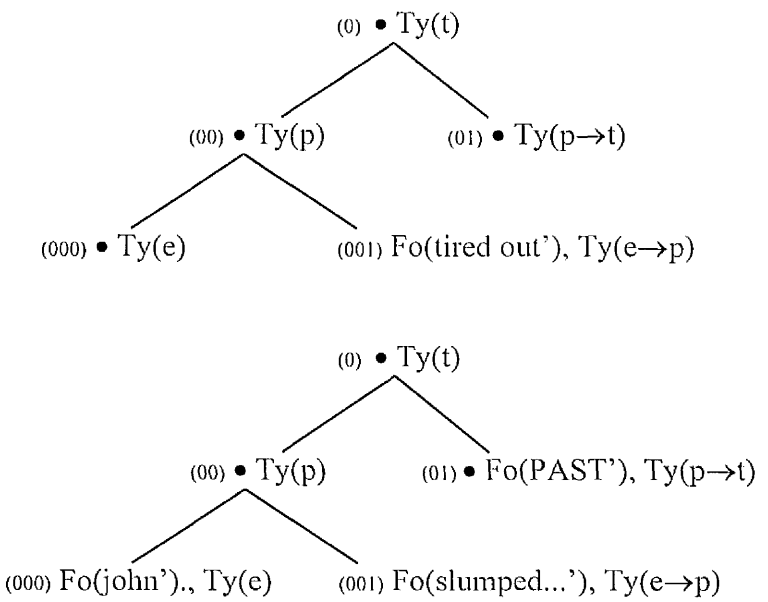
What information does the system receive now? Now there is an intonational break. This requires that a new tree of type (t) is set up. The only way in which trees can be related is through <LINK>. The situation is as illustrated in (135).

(135)



Here there are two trees, which are unconnected; moreover the first tree is incomplete, as it still has requirements to do. Processing the rest of the lexical input will yield a well-formed structure for the second tree, but the utterance as a whole will remain ill-formed as there is no connection between the trees and there are still outstanding requirements on the first tree. This is shown in (136).

(136)



In terms of what could resolve the situation, the solution is straightforward; as in the other cases of p-predicates used parenthetically, the <LINK> operation would unify the trees and satisfy the requirement for an entity at location (000) in the first tree. However, at no point in the derivation was it actually the case that the right conditions existed for the application of the <LINK> rule. There is a tension apparent between <LINK> as an update function and <LINK> as a

well-formedness requirement. The structure derived for (131) ought to be able to create a well-formed <LINK> structure; the question is how this is achieved.

There are three possible approaches to resolving this problem.

i. The initial p-predicate input could be specified as underlocated, and remain as a requirement on the tree; the <LINK> operation is launched subsequently, and the p-predicate is duly used in the L-tree.

ii. A metavariable could be supplied as a placeholder, where the metavariable is subsequently replaced by the node description held in common. The <LINK> operation could then either apply retroactively or could create the second tree as an L-tree.

iii. The <LINK> operation could be broken down into its constituent parts; so that there are separate operations of allowing for the specification of a new tree requirement and for guaranteeing that some update function occurs to derive a well-formed connection between the two trees.

In the rest of this section I explain why (i) and (ii) should be rejected. I adopt (iii) and argue that this is also the best way to characterise the <LINK> operation as generally applied in the case of p-predicates. This is developed in 6.5.

6.4.2.2. Underlocating the Predicate

Underlocation is available as a general strategy for processing fronted constituents in LDSNL. That is to say, any initial constituent can be assigned an underspecified tree node location, such that it has to be used subsequently in the process of tree development. This strategy is outlined in Kempson and Meyer Viol (1998).

What would happen if this strategy were to be applied in the case of parenthetical constituents? Such an approach *could* be implemented. In this case the definition of underspecification would have to be include the relation <D>, where <D> is a child relation that can hold across <LINK> relations. The predicate would be carried down until such time as a <LINK> operation is

launched; the predicate would then be transferred into the L-tree, along with the requisite description for the shared node of type (e).

The effect of this would be to have the parenthetical as a regular fronted construction. This is problematic because of the way that the current phenomenon differs from other cases of 'fronting'. The regular cases of fronting are ultimately located within the tree where they are originally specified. There is no intonational break.[24] Moreover, they are integral to the requirements of that tree building operation. In the present case the predicate is additional to the requirements of the tree. As the tree develops, there is no reason why this p-predicate should not be incorporated as soon as the requirement of type (e→p) is induced by the subject; in such an event it is not clear what would happen to the next predicate scanned. If the predicate is held as underspecified until a well formed tree has been developed, what this means is that the <LINK> structure will not be launched until the rest of the processing task has been completed. If this were the case there would be nothing to account for the fact that utterance initial parentheticals can only be used with the subject.

6.4.2.3. The Metavariable Approach

The result of scanning the lexical information in the initial tree still leaves that tree overall with a requirement of type (e). A possible solution would be to supply a metavariable in this environment, to 'assume' that there is some node description there, as in some form of PRO element. Given such a characterisation of the initial tree constructed from the parenthetical input, then there are three subsequent options for deriving a well-formed structure overall. Either (i) the subsequent tree would have to be labeled as the L-tree, or (ii) the tree provided by the parenthetical would have to be identified as the L-tree from the start or (iii) some form of inverse <LINK> rule would have to be adopted, of the sort defined in Kempson, Meyer Viol & Gabbay (in prep.) for head-final relatives in Japanese.

---

[24] Indeed, Kempson & Meyer Viol (1998) provides a separate account for elements considered outside the main processing task.

In the first case the <LINK> rule as formulated in (128) could apply, but the main tree, ie the tree with lexically supplied tense, would inherit a description as the L-tree. Hitherto, it has always been the case that the tree which was not exhaustively specified by lexically supplied information was characterised as the L-tree. The L-tree should be dependent, unable to exist except by contracting some relation with a fully specified tree. To maintain such a characterisation means abandoning the first of the three possibilities.

The second possibility requires that an L-tree be built from the start for the parenthetical. This is in itself stipulatory, but could be associated by rule with p-predicates in such a processing environment. It is still necessary to reconfigure the <LINK> rule such that it allows a mapping from an L-tree onto what will be the main tree. (137) gives a possible formulation.

(137)

$$\frac{[_L\ [_n\ Fo(U), Ty(e))]] \bullet Ty(t)]}{[_L\ [_n\ Fo(U), Ty(e))]] \bullet Ty(t)]\ ,\ [\ \bullet Ty(t);\ [_*\ Fo(U), Ty(e)]]}$$

This states that from an L-tree it is possible to launch a new tree of requirement type (t), further specified with the requirement that located somewhere in the subsequent development of that tree is a node identical with one of type (e) in the L-tree. However, there is no node description associated with the initial occurrence of that node; it is in fact a metavariable. This then has to identify with a node that is subsequently created in the main tree. The technical details of this could be developed, but it is clear that this provides a very different analysis from the other cases. Nor does this approach reflect the fact that the 'metavariable' does not really exist as a distinct node until such time as it has been identified.

This is similar to the third approach: to adopt an inverse <LINK> rule of the kind developed for head final relative clauses in Japanese in Kempson, Meyer Viol & Gabbay (in prep.).

(138) <LINK> Rule for Relative Clauses in Japanese (Provisional)

$$\frac{[_m \ldots \bullet Ty(t)]}{[_m \ldots [_{<d>0} Fo(U_{pro}, Ty(e)] \bullet Ty(t), <L^{-1}> [_* \bullet Fo(U_{pro}), Ty(e)], [_{n..} \bullet Ty(t)]}$$

This starts from a root node with requirement of type (t); the subject node is built, annotated with the description of type (e) and a metavariable as the formula; the requirement is further specified that a <LINK> relation be derived from another tree where some node will be located which identifies with the metavariable. A new tree task is then launched with requirement of type (t).

This rule defines the subject (of the L-tree) as the entity which has to be the node shared between the L-tree and the main tree. The consequence of this is that the U_{pro} cannot be identified until the <LINK> operation has taken place, and is thus a case of anaphoric linkage in the 'wrong' direction. Since <LINK> is from type (t) onto a node of type (e), any node of type (e) will satisfy it, so this can operate in the case of unfixed nodes as well as subjects.

In the present case, however, such a rule seems unmotivated. Whereas in Japanese this rule is the general way to build relative clauses, the utterance initial parenthetical is the only case in English for which it would apply. It could be argued that the same strategies should be characteristic of the parser, so that these are innately available in the course of language acquisition and then only a certain subset become 'grammaticalised'. In this case it should not be surprising that the same form of <LINK> rule is available both in English and Japanese. However, (138) is a specific rule instantiation rather than a general strategy; it plausibly stems from particular properties of Japanese which are not true of English. Japanese is head final and English is not. Japanese freely allows for null arguments with no lexical information to identify them;[25] pronouns are generally compulsory in English. Moreover, this rule is associated with a relativiser in Japanese, while in English there is no lexical item to encode it. Finally, for Japanese this builds integrated L-trees, whereas in English the L-tree derived from a parenthetical, the only case where this rule would apply, has to be non-integrated.

[25] Therefore, the metavariable could be supplied by the verb rather than having to be written into the <LINK> rule.

There are also properties of the approach itself which I believe make it undesirable as an account of utterance initial parentheticals. In no other case of building L-structures for p-predicates is it necessary to invoke some PRO element. In general the shared element is created by the <LINK> operation itself without having to be additionally specified. This use of U_{pro} seems extremely stipulatory in two areas. On the one hand, while the general approach of invoking a metavariable is not unusual in the terms of the present framework, in all other cases the metavariable is supplied by lexical material. On the other hand, in all other cases of the <LINK> operation in English, when the node identified as the shared node is carried into the other tree, it has some lexically supplied content. That is to say, the node in the main tree has been annotated with a description by a lexical operation at the point at which the <LINK> rule applies. It is unclear what it means to impose a requirement for identity with a PRO element where that PRO element has not yet been identified.

A further problem for both of the <LINK> rules suggested above is that they lose the incrementality of the structure building operation as they enforce an operation whereby the root node of the new tree is built before the processing of the first tree has been completed. Note that these rules have to apply before the predicate (ie the initial lexical input of the string) has been scanned. The effect is that the pointer would be moved into the new (main) tree; so at the point at which the parenthetical predicate is scanned, the pointer is already in the next tree, and therefore the lexical update cannot apply. There is no way to return the pointer to the L-tree without further stipulation.

There are, then, serious objections to both of these approaches. Furthermore, the effect is to separate the utterance initial case from the other cases of building L-trees from parenthetical p-predicates (and from all other instantiations of the <LINK> operation in English). Building a metavariable into the process detracts from the general characterisation of how the <LINK> operation works. There seems to be no motivation for distinguishing the utterance initial cases in this way. My aim is to identify the general aspects of the <LINK> operation. Therefore, rather than stipulating the existence of a metavariable, I reconsider the dynamics of the <LINK> operation.

6.5. Reconfiguring <LINK>

The approach that I adopt to the problem of building structure for utterance initial parentheticals involves subdividing the <LINK> operation into two distinct processes. This not only gives the correct characterisation of utterance initial parentheticals, it will also allow a unified solution to the question of building L-trees for p-predicates.

My concern is to maintain as uniform an approach as possible to the different instantiations of p-predicates constructing L-trees. The particular problem of the utterance initial case is to build structure when a premise is lacking; this is, in fact, a stark example of a problem which also exists in the other cases. In the previous chapter, and for the other parenthetical cases, I noted that there was a choice as to the node identified as the shared one, and that identification of this might depend on the predicate in the L-tree. So in fact, in these cases too there is the problem that for structure to be built, there is the tacit assumption that the node has already been identified, but the identification of that node depends on returning to the main tree. Moreover, the way the formulation of the <LINK> rule is developed assumes that the node has already been identified.

The account I now develop also proves more explanatory of the way in which <LINK> is available as an option in the structure building process. In 5.2.1. I raised the problem with regard to <LINK> and discontinuous input of whether the <LINK> rule should be applied in order to allow structure to be derived for the adjective; or whether structure is built for the adjective and then the <LINK> operation guarantees a well-formed connection to the main tree. What seems problematic about the conceptualisation of the process hitherto is the way that the <LINK> operation and the search for an antecedent combine.

The <LINK> rule states that there is a certain node that is carried over into the L-tree; however, as I have made clear, it may be the case that this node is not identified at the point at which the L-tree is built. There is a tension here between the purpose of <LINK> as an update function, and the purpose of <LINK> as a condition of well-formedness. Two steps constitute the <LINK> operation: (i) the establishment of the task of constructing a new tree and (ii)

the imposition of a requirement of a shared node. I now examine how the link operation can be separated into these two parts.[26]

In the most general tree-building terms it is always possible to launch a new task (according to the sort of input that is coming in). To guarantee structural well-formedness overall, the premises all have to be used and the trees must combine to form some kind of coherent structure, that is to say, there has to be some connection between the trees.[27] With regard to the present case, the launching of a new tree structure is a freely available option in the system, but once that step has been taken, what is compulsory to guarantee the well-formedness of the utterance is that the two trees are connected by the <LINK> operation, ie they must share a node. The specification of <LINK> as update then becomes an update function over two trees.

6.5.1 Establishing a New Tree Goal

If a new goal is established which cannot be fulfilled within the current requirements, then this establishes the root of a new tree.[28] This is effected in terms of tree update by the rule given in (139).[29]

(139) Additional Tree Construction Rule

$$\frac{[_{<u>*} \bullet Ty(x)]}{[_{<u>*} \bullet Ty(x)], [_0 \bullet Ty(x)]}$$

where x ranges over the set of well-formed type expressions.

---

[26] The general approach to specifying rules in Kempson, Meyer-Viol & Gabbay (in prep.) is that a form is given which in fact combines the various steps necessary to achieve the correct result.

[27] As to how else a well-formed connection might be achieved between two trees, I have nothing to say.

[28] Recall that every tree represents a proposition.

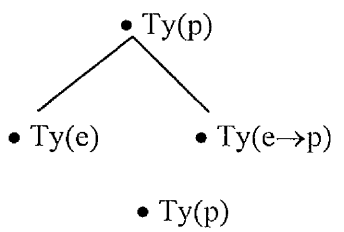
[29] This could also be written in the form given in (i).

(i) $$\frac{[_n \bullet <u>* Ty(x)]}{[_n \bullet <u>* Ty(x)], [_0 \bullet Ty(x)]}$$

This is defined such that the new tree launched can be of a type no higher than the currently existing goal in the tree from which the operation is launched. What (139) states is that at some node location in a tree, where there is a requirement of type (x) either at that node, or at some node above that node in the same tree, a new tree can be launched, annotated on the root node with the requirement of type (x). The Additional Tree Construction Rule is independently necessary in the system, for example to allow co-ordination.[30]

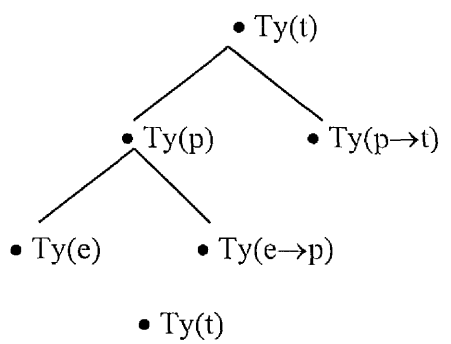
For example, in (140) the new tree shown could have been launched from any of the nodes.

(140)



Similarly in (141).

(141)



In (141) a new tree annotated with the requirement of type (p) might equally have been launched, or for that matter of type (e), but this depends on the position of the pointer. A new tree with the requirement of type (p) cannot be launched from a node of type (t) for example.

It may seem that this allows an extremely unconstrained system, but the proliferation of structure is constrained by the general well-formedness requirements of the system. Any additional tree that is built is subject to the

[30] It also allows the system to derive structure for cases of interpolation such as (28) and (29) above.

general constraint that all the lexically supplied information is used and that no requirements are left to do at the end of the process, and that the all premises combine in a process of compilation. The way that new tree rule has been set up, the type specification on it can be decided according to general information in the parsing process; for example, utterance finally L-trees may still be in the same domain, whereas with the parentheticals a new processing task is established. This is guided by intonation.

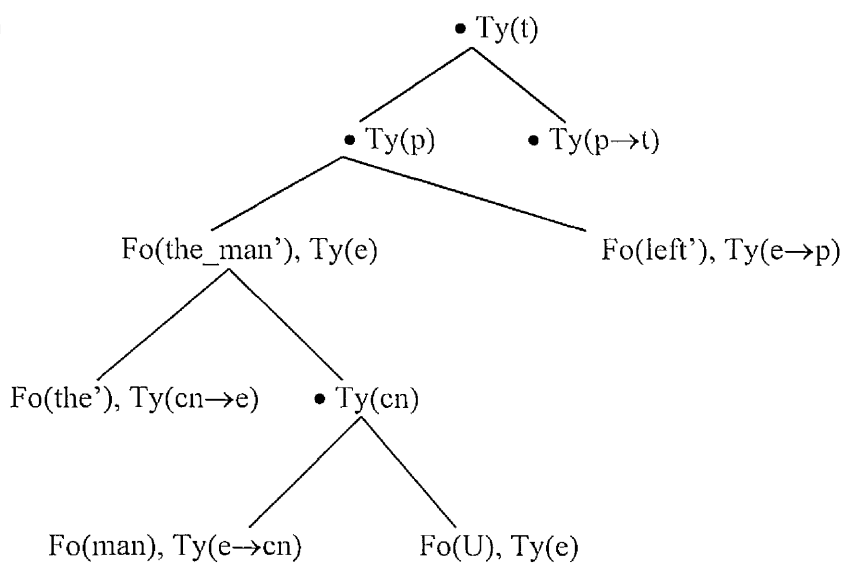
6.5.2. Establishing a <LINK> Relation

What the <LINK> operation has to do now is combine two trees such that they form a well-formed whole. This maintains the original insight that there is the requirement for a shared node, but integrates this more with the general process of structure building. To see what sort of input there might be to this new style of <LINK> operation, and how the rule itself should be characterised, I consider the how a derivation proceeds in light of the Additional Tree Construction Rule.

I first consider the discontinuous input case which I discussed in the previous chapter, given here as (142).

(142) the man left angry

(143)



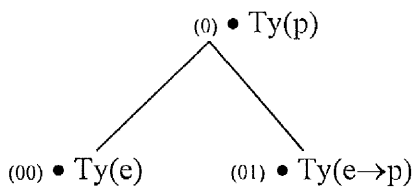
At the end of parsing the main part of the utterance, the structure built is as in (143). What happens now is that according to the rule stated above, a new tree can be launched. This simply gives the representation in (144)

(144) • Ty(p)

At this point there is no specification that this is a <LINK> tree.

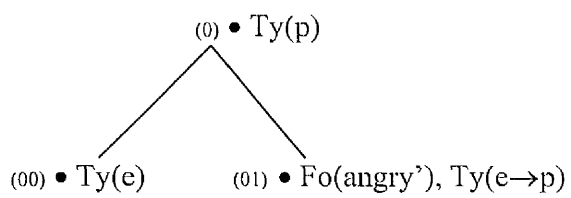
What happens next is that the requirement is broken down in the familiar way, to give the output in (145)

(145)



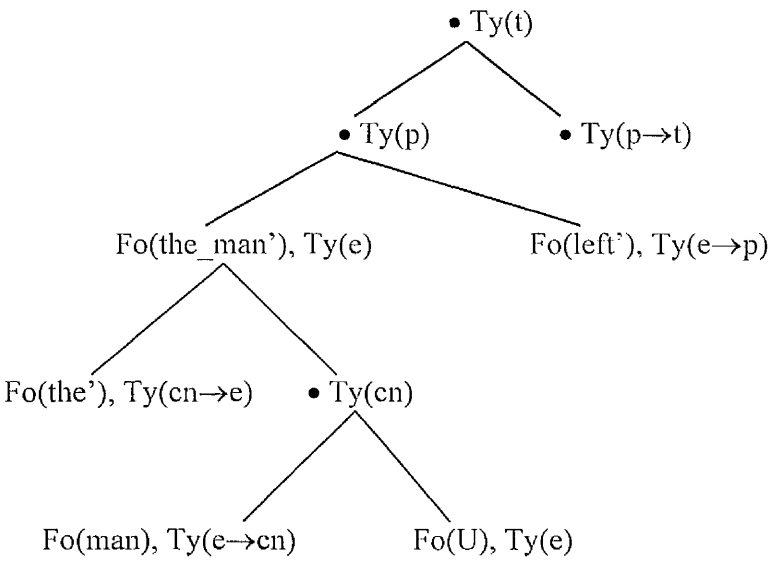
Processing continues; the lexical input is scanned and the structure in (146) is derived.

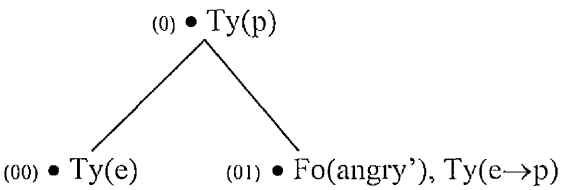
(146)



At this point there is no further lexical input, but the tree is not yet well-formed: the requirement for type (e) at tree node (00) remains unfulfilled. The overall state of tree growth for the utterance is shown in (147).

(147)





The <LINK> operation, as it has previously been characterised, requires there to be a shared node between the two trees; in this example, there is an outstanding requirement for an entity at (00).[31]

The pointer returns to the main tree; at some location in that tree applying the <LINK> rule given in (148) would result in a node being carried over into the other tree, which would then be labelled as an L-tree; (the node where the rule applies is the one that is carried over).

(148) Provisional Characterisation of Link Rule (i)

$$\frac{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)], \, [\, \bullet \, Ty(p), \, [_{<d>} \bullet \, Ty(e)] \,]}{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)], [_L \, \bullet \, Ty(p), \, [_{<d>} \, Fo(\alpha), \, Ty(e))]]}$$

This rule states that given a tree with some node of type (e) and given some tree with the requirement of type (e) in the subject position, then that node can be

[31] Note that this situation is analogous to that of the utterance initial parenthetical described in 6.4.2..

taken over to fulfil the requirement. The tree that has an outstanding requirement for an entity can have this requirement satisfied by a node description from the other tree. It thus becomes an L-tree. This rule successfully characterises the particular case under discussion, but it is too specific to cover all the cases of the <LINK> operation.

To arrive at the correct formulation of the <LINK> rule, it is necessary to take into account other cases. The way that the rule is set up in (149), it can apply only to the cases of p-predicates I have just been discussing. The p-predicates are different from relatives where there is a relativiser which performs the update, but the instantiation in (149) is so specific it would allow the rule to apply only in those cases where there is no temporal label supplied on the L-tree. However, the Additional Tree Construction Rule does not specify that any additional tree has to be of type (p), so, similarly, this level of generality should be built into the <LINK> rule. Indeed, the parenthetical cases, though not lexically supplied with tense, have been characterised as being of type (t) inasmuch as they do form a separate inferential unit. So (149) can be recast as (150).

(150) Provisional Characterisation of Link Rule (ii)

$$\frac{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)], \; [\, \bullet \, Ty(x), \; [_{<d>} \bullet \, Ty(e)]}{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)], [_L \bullet \, Ty(x), \; [_{<d>} \, Fo(\alpha), \; Ty(e))]]}$$

This maintains the condition that the tree which supplies the lexical information characterising the shared node has the overall requirement that it is a tree of type (t). The L-tree can be of any type. This allows for L-trees then to be marked with a temporal location distinct from that of the main tree.

The L-trees derived from p-predicates all have the property that the shared node is used as a subject, and this is reflected in (150). Consider, though, cases where the L-tree is tensed but there is no complementiser, such as (151).

(151) the man I like arrived.

To capture these cases, the tree location of the shared node can be described as underspecified. This gives the form in (152).

(152) Provisional Characterisation of Link Rule (iii)

$$\frac{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)], \, [\, \bullet \, Ty(x); \, [_* \bullet \, Ty(e)]]}{[_m \ldots\ldots\ldots [_n \, Fo(\alpha), Ty(e)] \ldots \bullet \, Ty(t)], [\, _L \bullet \, Ty(x); \, [_* \, Fo(\alpha), \, Ty(e))]]}$$

Such a <LINK> rule could then apply in those cases where a complementiser-less relative is marked with tense. The star notation indicates that the nodal description holds either at the current node or at a child of the current node, or at a child of a child of the current node recursively (cf 2.1.3.). This weaker characterisation indicates merely that this node is located somewhere within the current tree.

(152) gives the correct form of the <LINK> rule to cover the various permutations of tree that can serve as input to the rule. However, as it stands, this rule still requires that the pointer be located at the node to be transferred, node (n), in order for the rule to apply. In fact, as discussed in chapter five, the restriction on identifying the node to be shared is such that any node that is a child of the current pointer location can be identified as the node to be shared. This is straightforward to write into the rule using the tree node description language. The way to write this is shown in (153).

(153) $[[_{n \vee <d>} \, Fo(\alpha), \, Ty(e))]]$

This states that a node description holds either at the current node or at a child of the current node. I use '$\aleph$' to indicate this description, rewriting (153) as (154).[32]

(154) $[[_{\aleph} \, Fo(\alpha), \, Ty(e))]]$

[32] In contrast to the star description <*>, this is *not* recursive.

What this property reflects is the way that the tree has been built up. The description on a child node, after application of Completion, is annotated on the parent as holding at the child. This explains, then, why such information is accessible.

(155) <LINK> Rule (Revised)

$$\frac{[_m \ldots [_n\ Fo(\beta), Ty(y)], [_N\ Fo(\alpha), Ty(e))] \ldots \bullet\ Ty(t)],\ [\ \bullet\ Ty(x);\ [_*\ \bullet\ Ty(e)]]}{[_m \ldots [_n\ Fo(\beta), Ty(y)] \ldots \bullet\ Ty(t)],\ [_L\ \bullet\ Ty(x);\ [_*\ Fo(\alpha), Ty(e))]]}$$

This states: given a tree with the requirement of type (t), and given another tree with the requirement of type (e) holding somewhere within that tree, and given that the pointer is in the first tree at node location (n); then the pair of trees can be updated such that a node description holding either at the pointer location or at a child of the pointer location is used to satisfy the requirement of type (e) in the second tree. This second tree is then labelled as a <LINK> tree.[33] The rule is now a general characterisation of the way in which the update function performed by <LINK> is realised.[34] This rule maintains the notion that the pointer has to be at a certain location in order for the rule to apply. Where does application of this rule leave the pointer? The answer is, in the <LINK> tree; note that that is where some operation has been performed on a node. When the <LINK> tree is completed it returns to the point where it came from, ie wherever it was in the original tree that the <LINK> operation applied.

How does all this work in the actual process of structure building?

6.5.3. Structure Building for L-trees

I now examine the way in which the process operates to derive <LINK> structures. I discuss first a generalised case to show *how* the operation works; I then turn my attention to when the <LINK> operation should occur. I consider the building of structure for the example in (156).

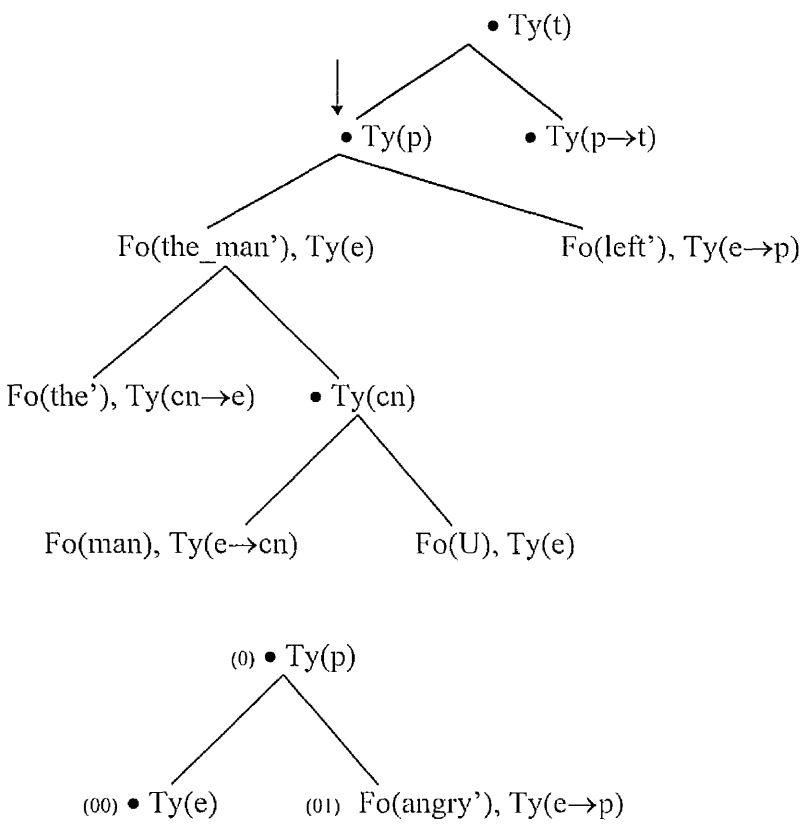
[33] By using the term 'second tree' I do not mean to indicate any ordering in terms of deriving the trees. The ordering of the two trees is irrelevant to the application of the rule.

[34] The <LINK> rule given here imposes the requirement that the shared node be of type (e). Cases where the shared node is not of type (e) are discussed in the next chapter.

(156) the man left angry.

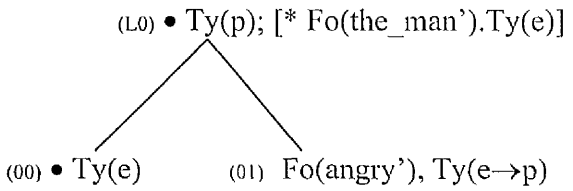
Assume that scanning of all the lexical premises takes place; the Additional Tree Construction Rule allows the launching of a new tree task, building of a tree for the predicate 'angry'. However, that tree is not yet well formed, as it still has the requirement for a node of type (e). The pointer returns to the initial tree. (157) shows the overall structure derived immediately prior to application of the <LINK> rule. One tree has an outstanding requirement of type (e); when the pointer reaches the node with requirement of type (p), indicated by an arrow, the conditions are met for the application of the <LINK> rule.

(157)



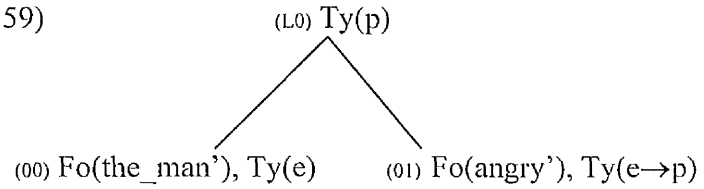
The effect of the <LINK> rule applying is to update the P-predicate tree to give the structure in (158).

(158)



Here a node has been brought over from the other tree, and assigned an underspecified location, and the root node of this tree is labelled as a <LINK> tree. The underlocated node is duly annotated in the subject position, location (00), and the tree can compile:

(159)



The pointer then returns to where it left off from in the original tree, and compilation occurs there.

The question remains as to when exactly the <LINK> operation should apply. What the <LINK> rule states in essence is that there is a completed e-node in one tree and a requirement for an e-node in the other tree. This may be the case even before the predicate is scanned. The Rules of Introduction and Prediction allow the initial tree requirements to be built. So immediately this has happened, the <LINK> rule could apply to give the shared node as the subject. The above example gives the generalised case. However, there is no reason to delay any operation if the conditions are right for it to apply. There are two options available, then, for when the <LINK> rule should apply: either it applies immediately that the requirement for a node of type 'e' is identified; or it does not apply until such time as the lexical material in the L-tree has been processed.[35]

After the first tree has been built and all the requirements there have been satisfied, the new tree task is launched according to the Additional Tree

[35] As shown in the generalised case above.

Construction Rule. Given the strong preference in English for identification of the subject, it may be that in fact that the <LINK> rule does apply immediately. How would that work?

This strategy would apparently require that the pointer move back to the main tree, if application of the Additional Tree Construction Rule moves the pointer to the new tree. An alternative approach is that at the point at which the new tree is built the pointer is in a sense in both trees; it will only move when more structure is built. The requirement for a subject can be identified automatically by Introduction but this does not mean that the node has to be built; the <LINK> node can then apply to supply the requisite node, so that the shared node is identified at the inception of the L-tree. This all occurs without the pointer having been advanced from its location, and this node is then subsequently located as the subject. This is illustrated below in (160); the pointer in the main tree is at the node of type (p). In the new tree, the requirement for a subject is immediately derivable as a requirement on a child node.

(160) $\bullet$ Ty(p), $[_{<d>} \bullet$ Ty(e)]

At this point, the <LINK> rule can apply and the shared node can immediately be instantiated.[36]

(161) $\bullet$ Ty(p), $[_{<d>}$ Fo(α), Ty(e)]

Note that the location of the pointer at the time at which the Additional Tree Construction Rule is launched will determine what is accessible. The overall result is that underspecification is written into the structure building process; the point to resolve is at what stage to launch the new tree. Thus, the shared node does not have to be identified in order to launch the new tree task.

[36] This still preserves the insight that what happens to the shared node in the new tree depends on the development of that tree.

Adopting this approach, I suggest the following possibilities. What happens in the case where there is no other NP in the initial tree, as in (162), is that the pointer is at node (p) when the new tree is launched; and application of <LINK> identifies the only possible NP. This may happen before the predicate is processed for the new tree.

(162) Janice drank the beer giggling.

In the case of (163), where there is an object, then the new tree can be launched either from the (e→p), in which case that object will be accessible; or from the (p) node, in which case only the subject will be accessible. This is a choice that has to be made on-line, and is presumably guided by intonation.

(163) Billy ate the chicken naked.

The 'safest' choice is to remain at the (e→p) node, so that the object will remain as a possible option. In this case, I suggest what happens is that the predicate in the new tree may be scanned before application of the <LINK> rule.[37]

The above discussion is intended to illustrate the options that are available; definitive conclusions about the order of application of operations must await further research.[38] The strength of the approach presented is that it allows new tree structure to be derived before the shared node description is identified and that different types of cases may be captured by differences in where the rules apply, rather than by having to postulate separate rules.

---

[37] This avoids the problem of structural reconfiguration.

[38] Cases such as (i) might suggest that the linked node has to be identified in order to resolve pronominal choices on-line. These require an anaphoric choice on-line. However, it is not the case that these have to identify with the subject of the L-tree, as (ii)-(iv) show.

(i) the man$_i$ left angry that his$_i$ mother had ignored him$_i$.
(ii) John sent his son$_i$ to live with relatives abroad fearing for his$_i$ safety.
(iii) with her$_i$ permission, Sandy took Cherie's$_i$ car.
(iv) the man attacked the policeman$_i$ angry that his$_i$ colleague had arrested him.

I take these to involve a general anaphoric choice from the available referents.

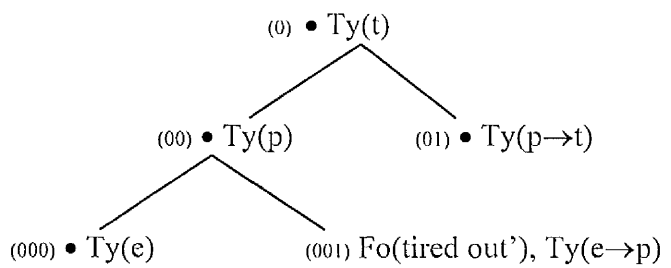
6.5.4. Building Structure for Utterance Initial Parentheticals

I now return to the case of utterance initial parentheticals which were the start of the problem. In this case, the predicate has to be scanned before the shared node can be identified: the predicate occurs first. The example is repeated as (164).

(164) tired out, John slumped in the chair.

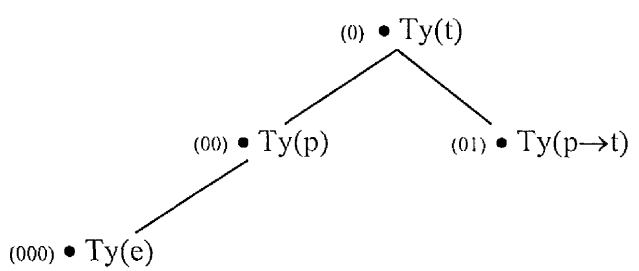
Processing the parenthetical gives the structure in (165).

(165)



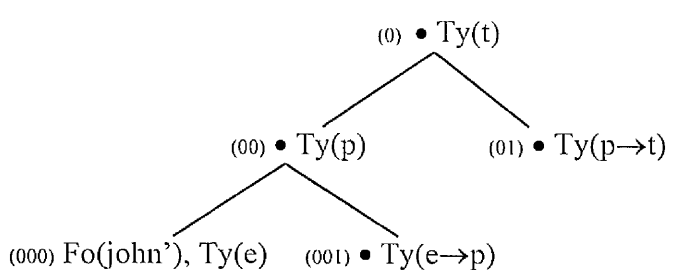
Application of the Additional Tree Construction Rule allows the launching of a new task with requirement type (t), which is broken down into subconstituent requirements, as is shown in (166).

(166)



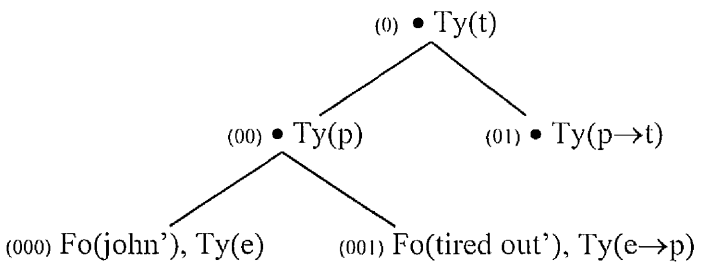
The next lexical item to be scanned is the noun phrase 'John', the lexical entry for which causes this tree to be updated to (167).

(167)



At this point there is a requirement of type (e) in one tree, and a description of type (e) in another tree. The conditions are right for the <LINK> rule to apply. This updates the tree created by the parenthetical; first the 'john' node is carried over to this tree, as underlocated, and the tree is labelled as an L-tree, then the underlocation is resolved in the subject position, to give the result shown in (168).

(168)



This then constitutes a well-formed <LINK> tree (subject to satisfaction of the requirement for a temporal location). The pointer returns to the main tree and the process continues as normal to build the main tree for the utterance.

The reason that the <LINK> operation applies when it does is the general necessity to satisfy requirements as soon as the right conditions prevail. The <LINK> operation takes place as soon as it can. This explains why only the subject of the main clause can provide the shared node for the L-tree. Similarly, with other parentheticals, the <LINK> rule has to take place to guarantee that the tree the parenthetical maps onto is well formed; this has to occur as soon as possible. No node built in the main tree after the parenthetical has occurred can be the shared node.

What this establishes, then, is the general tree building strategy of being able to start a new tree, and then a specific instantiation of the way that trees are connected.

6.7 Conclusion

In this chapter I have examined the parenthetical use of the p-predicates. Parentheticals are problematic for syntactic theories because of the way that they

seem not to form an integrated part of the sentence. Nonetheless, the basic predicates, used parenthetically, have to contract a semantic relation with the main sentence in order to be interpreted. In LDSNL the apparent independence is explained by the fact that the parenthetical is located within a separate processing domain of type (t). However, the well formedness of the tree that is built for the parenthetical is guaranteed by its contracting a <LINK> relation with the main tree. In this respect, parentheticals do not differ from 'sentence internal' <LINK> structures; the same mechanisms are used to build the tree structure. To identify the correct structure-building mechanisms I re-examined the <LINK> operation. I broke this down into two steps: launching another tree with a new goal to do; and connecting the two trees so that there is a shared node. The tree containing the lexical origin of this shared node is the main tree; the tree inheriting an entity otherwise lacking is labelled the L-tree. This account provides a more detailed means of examining the way in which the shared node is identified.

The <LINK> relation indicates a relation between trees which go to make up the structure of an utterance. It does not combine them into a single tree, however. That is to say, although this is a relation in terms of being defined over trees, it is not the case that the root nodes of these trees are combined by function application.

In the final chapter I give an overview of the thesis; the results obtained and the wider implications, considering the overall picture and how <LINK> structures might be incorporated into pragmatic processing.

Chapter Seven

Conclusions:

<LINK> Reviewed

7.1. Summary

I start this chapter by giving an overview of the account I have presented. I started by introducing the formal machinery of the framework, LDSNL, which I used in the rest of the work. Utterance interpretation involves building semantic trees. Tree building is licensed by transition operations, and by update instructions provided by lexical items. The tree node description language allows for underspecification. Crucial to the development of the rest of the thesis were the explicit ways that the framework allows to represent the *stages* of the structure building process. Any node location can be annotated with requirements which still have to be done at that location, and with descriptions of information that holds there. The modalities of the tree node description language allow that at any node, information pertaining to another node can be annotated (subject to this information having been derived).

I introduced the mechanism which forms the basis of inquiry for the rest of the work. This is the <LINK> operation (Gabbay and Kempson (1992)). The basic insight of <LINK> is that two (or more) trees can be conjoined by dint of a node description being held in common between them. Such a connection is derived dynamically; the <LINK> operation provides a transition from one tree onto another and imposes a requirement of a shared node description.

I ultimately separate the <LINK> operation into its two constituent operations, arguing that these occur as distinct steps: the launching of a new tree task, and the imposition of the requirement for a shared node. The launching of a new tree is always a freely available option; successfully identifying a shared node guarantees a <LINK> relation between two trees. I arrived at this characterisation after examining the way that <LINK> could be developed to

cover a wider range of cases. At each stage, although it has been necessary to modify the exact formulation of the <LINK> rule, the basic operation is consistent: a new tree is launched which relies on another tree to derive a node description.

I have explored three lines of development for the <LINK> operation, extending its empirical coverage accordingly.
These were:

i. non-tensed predicates in the L-tree;
ii. non-continuity between the occurrence of a node in the main tree and the transfer of that node into an L-tree;
iii. non-integration of the L-tree into a processing task of the main tree.

Kempson, Meyer Viol & Gabbay (in prep.) notes that it is not necessary to have lexical material in the L-tree to instantiate the shared node. That is to say, it is possible to carry over the information to be annotated as a node description rather than as a requirement. This means that the node description does occur somewhere within the L-tree; the unknown is where in that tree the node is located. This is characteristic of relative clauses without a complementiser. The identification of this property opened up the possibility of applying the <LINK> operation to build structure for 'reduced' relatives.

I identified the lexical items that can be used to form reduced relative clauses, which correspond to those that require copular support to form main clause predicates. To analyse reduced relatives involves the representation of non-tensed propositions. I introduced the type (p) for propositions which have no temporal location, and defined predicative lexical entries for these in terms of type (p). This allows for a straightforward analysis of reduced relatives using <LINK>. The difference with these is that the L-tree that is built is of type (p).

The next challenge is how <LINK> can be used to derive structures from discontinuous input. The way that <LINK> is set up has the effect that it launches a new tree task from a certain node location and carries the description annotated on *that* node location into the L-tree. For extraposed relatives and

adjunct predicates, the node description which is shared between the two trees is *not* that on the node from which the operation is launched. This does, then, necessitate substantive development of the operation itself. I redefined the <LINK> operation such that it does not require identity between the node at which it applies and the node that is carried over into the new tree. In the case of extraposed relatives, the node which is to form the common element is identified by a search procedure in the main tree, where this procedure is initiated by the wh- relativiser. This node can either be the metavariable from within the common noun task, or it can be the whole of the e-task, depending on how the e-task has been processed. The identity of the shared node provides the basis for defining the terms 'restrictive' and 'non-restrictive' in the present framework as properties of the L-tree. When the shared node is the common noun metavariable, the L-tree is restrictive; when the shared node is the completed e-task, the L-tree is non-restrictive.

I thus analysed the way L-trees can be derived for p-predicates, and I analysed how L-trees can be built when the shared node is located earlier in the main tree development than the location from which the <LINK> operation is launched. From these two analyses the account of adjunct predication falls out naturally. Adjunct predication involves building L-trees of type (p) from discontinuous input. In these cases, the process of identifying the shared node is much more constrained, as it is restricted to child nodes of the node location at the point at which <LINK> is launched. This property reflects general facts about the system. When the pointer moves back up then that node description is annotated on the parent node as holding at the child by the operation of Completion. This means that any description of type (e) can provide suitable material to be the shared node. This also explains why these cases can never be restrictive; the common noun metavariable is not accessible at a child node of the pointer location.

Finally, I examined parentheticals. As with the cases already analysed, these require a shared node to be identified between the main tree and the L-tree built for the parenthetical predicate. However, parentheticals appear to be distinct from the syntactic structure of the main clause, hence the apparent anomaly they

present for traditional syntactic theory. In LDSNL, this can be explained by the relative processing domains: the L-tree lies outside the processing task which characterises the main tree. This requires that the parenthetical itself establishes a proposition of type (t); here the temporal location of the proposition may be contextually supplied. The process of assigning structure for parentheticals presents more sharply the issue latent in all cases of discontinuous input - is it necessary to have identified the node to be shared before the <LINK> operation can apply? I argue that it is not, and consequently divide the process of building L-trees into two parts. The Additional Tree Construction Rule allows a new tree task to be launched. The <LINK> Rule requires that a node description be identified which is held in common between the two trees. This preserves a notion of incrementality where the initial processing of the new tree task can be started immediately, allowing a uniform pair of operations to cover all the cases.

The properties associated with building <LINK> structures are that:

i) the building of a new tree can be launched on-line, and this does not have to make reference to previously established tree material.
ii) in order to guarantee a well-formed <LINK> relation, a node description has to be identified which is held in common between the two trees. The tree where this description originates is the main tree; the tree where it satisfies a requirement is the <LINK> tree.
iii) the properties of the L-tree are determined dynamically according to the way that structure has been built.

In the rest of this chapter I draw out the implications of the account I have presented, speculate about possible developments and indicate future lines of research.

7.2. A Unified Analysis

My aim has been to identify the properties common to all of the instantiations of the <LINK> operation. Although there may be differences in the trees derived, is it possible to give a uniform characterisation of the process?

Differences in the L-trees are caused by two factors: the processing environment and lexical input. The lexical input determines whether or not the L-tree is tensed, and whether or not it is derived with a relativiser;[1] these properties are connected, stemming from the fact that a tensed clause in English has to have a lexically marked subject.[2]

The presence of the relativiser allows a wider search space for the shared node, which opens up the possibility of restrictive L-trees being built for discontinuous input; but otherwise the relativiser does not essentially affect the <LINK> operation itself.[3]

In the last chapter I characterised <LINK> as being made up of two separate operations, launching a new tree task, and sharing a node description. I repeat below the characterisations I gave for these.

(1) Additional Tree Construction Rule

$$\frac{[_{<u>*} \bullet Ty(x)]}{[_{<u>*} \bullet Ty(x)], [_0 \bullet Ty(x)]}$$

(2) <LINK> Rule (Final Version)

$$\frac{[_m \cdots [_n Fo(\beta), Ty(y)], [_N Fo(\alpha), Ty(e))] \ldots \bullet Ty(t)], [\ \bullet Ty(x); [_* \bullet Ty(e)]]}{[_m \cdots [_n Fo(\beta), Ty(y)] \ldots \bullet Ty(t)], [_L \bullet Ty(x); [_* Fo(\alpha), Ty(e))]]}$$

[1] Although the relativiser may not be represented in the final tree, it will appear in the intermediate steps of structure building.

[2] Hence the acceptability of tensed restrictive relatives where the linked node is not in subject position:

(i) the man I like
(ii) the chair Sue sat on

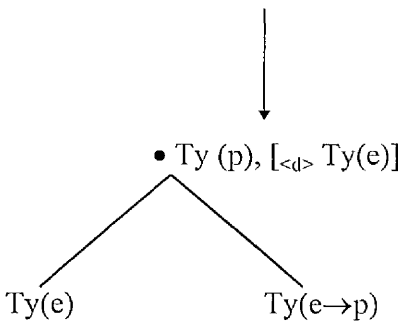
[3] I leave it as an open question whether or not some parts of the <LINK> operation should be encoded in the relativiser.

These were developed to capture all the possible cases; but is it correct to use this as a characterisation of all the cases?

In fact, I argue that this does have wider significance for the account. This analysis genuinely provides a uniform characterisation of the continuous and discontinuous cases. I have been concerned with issues of pointer movement and the impact of the location of the pointer on the building of L-trees. Recall that, originally, the way that the <LINK> operation was defined for cases of continuous input was that the node from where the operation is launched is the node that is carried over. This, by definition, is not the case for discontinuous input, hence the need to revise the way the operation was characterised.

In the case of discontinuous input, the pointer is at the p-task.[4] The node to be identified as the shared description could be at a child of the pointer location, reflecting the fact that the information concerning this child would be annotated on the parent node. This is shown in (3).

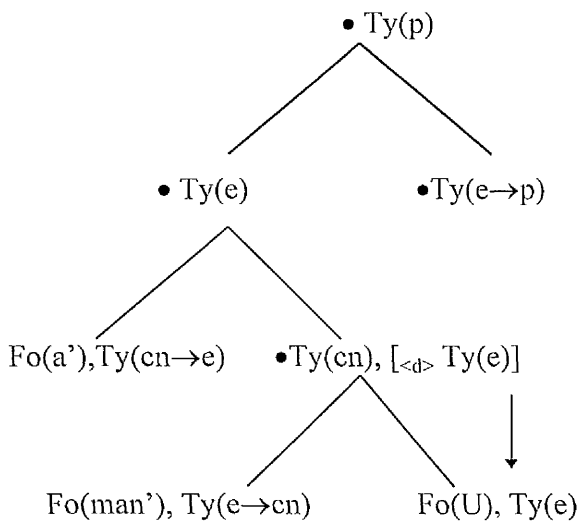
(3)



The case for continuous input, where the launch node is the node to be carried over, is shown in (4) .

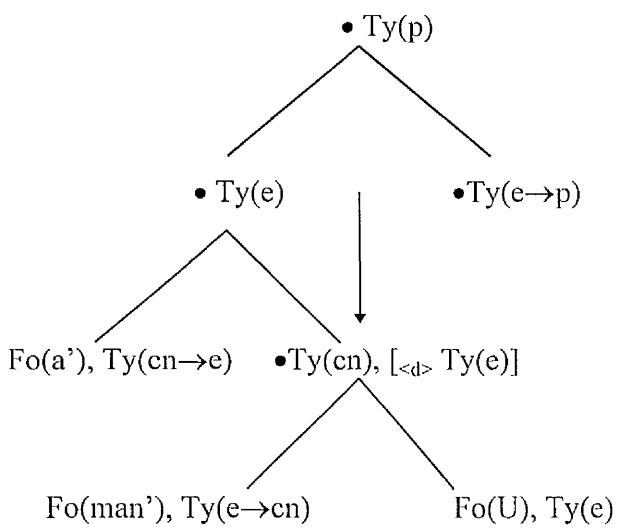
[4] Or the (e→p) task in the case of a transitive verb.

(4)



In fact, given the system now set up, the same analysis can be applied to both of these cases. The way I propose to analyse the case of continuous input is that the pointer has returned to the cn-task, and that it is from this node that the new tree is launched. In exactly the same way as with the discontinuous case, the node that is identified as the shared node is at a child node of the pointer location (note that the metavariable is of the required type). This is shown in (5).

(5)



This way of analysing the situation has two advantages. Firstly, it gives a single account of the two cases. Secondly, it preserves the general characterisation of scanning, completion and pointer movement. Lexical entries instantiate lexical procedures. When a node has been annotated with a description by a lexical

action, Completion should cause the pointer to return to the location from where the operation started. This is exactly what happens in the account I have given.

This opens a further line of inquiry with regard to operations within the e-task, which I sketch out as follows. The introduction of type (p) introduces the possibility of redefining the common noun and the determiner. Rather than having the type (cn), this type could be replaced by defining common nouns as being of type (p). The common noun would still contain a metavariable of type (e) but the predicate would be (e→p), and the overall type would be (p). The determiner could then be defined as type (p→e). This would bring down the number of types to the original number. As far as I can tell, this does not immediately have any adverse effects on the rest of the system.[5] What other advantages would it have?

This proposal would reduce the apparent anomalies when restrictive L-trees are conjoined with the common noun. In the case of an L-tree of type (p), the same type is involved. In the case of an L-tree of type (t) the discrepancy is less than with type (cn), as the only difference is in terms of being specified or not with a temporal location.

From this arises the possibility that, when launching a new tree before the existing one is complete, the same task is carried over into the new tree as characterises the location in the existing tree from where the new tree is launched.[6] That is to say, a more restricted way of defining the Additional Tree Construction Rule is that in any given processing domain, a new task can be launched only of the *same* type as the location at which the rule applies. In this case, in the case of continuous input, a new tree with requirement of type (p) would be launched, thus giving the same basic type. The tense information supplied in the case of a finite clause would then come after and be annotated on the L-tree, but not on the 'proposition' supplied by the common noun. This involves the building of a tree of type (t), when the requirement is of type (p). I

[5] A potential problem is the way that the subcategorisation requirements of the determiner could be satisfied by any proposition of type (p). The ramifications of this remain the subject of future research.

[6] The intonational break with parentheticals requires that these be of type (t) , however.

see nothing intrinsically wrong in this, as the requirement of type (p) is also satisfied.

The idea of defining common nouns as type (p) and the subsequent speculation remain to be examined in all their implications; I present them as possible lines of future development.

The major point is the reanalysis of <LINK> in terms of the pointer location. What this allows is that not only can the <LINK> rule be stated in general terms, but that the way the operation proceeds is exactly the same in all the integrated cases, both continuous and discontinuous. The node description held in common between the two trees is always a child of the pointer location.

7.3. <LINK> and Interpretation

My major concern has been the way that structure is built for L-trees. The advantage of the account that I have given is the ways in which it allows for the incorporation of different factors. I have distinguished between the node location from where the new tree is launched and the node location where the shared node is identified. This allows for the possibilities of restrictive L-trees formed from discontinuous input, and non-restrictive L-trees which are nonetheless integrated. These factors are all separately predicted to have effects on the interpretation. But what of the basic question of what it means for a tree to be an L-tree?

What do all the L-trees have in common? They all share the notion of dependency. In all cases, L-trees rely on some other tree for part of their interpretation. The logical implication of this in terms of current approach is that there should be some impact on the status of these trees as inferential vehicles. That is to say, the L-tree has to be of lesser importance in terms of the overall communicative objectives than the main tree. The L-tree cannot carry the main interpretive burden of the utterance, in other words it is not the main point of what the speaker is saying.

I believe that this will have to be defined in terms of some notion of main relevance for the set of propositions derived by the process of utterance

interpretation.[7] The main relevance is attached to that proposition within the utterance which the speaker intends to achieve the major contextual effects. In general this is identified with the node of type (t).

As a starting point, I suggest that any tree with a root node of type (t) which does not have any other root node above it will itself achieve contextual effects. Any tree that does not have a root node of type (t) is not required to achieve contextual effects independently. The contribution it makes has to be in combination with some other proposition. Note also that the main tree has all of its structure built on the basis of the lexical premises which supply the input to its building. An L-tree, in order to be well-formed, has the same requirement of having all its premises used and all of its requirements fulfilled, but in this case, the only way that this can be achieved is by 'acquiring' a node from elsewhere.

In truth-conditional terms <LINK> has been defined as mapping onto conjunction. However, what this means in the present framework is a processing notion - the two conjoined propositions have to be processed together. This ties in directly with Relevance Theoretic approaches to conjunction. Carston (1993, 1998) analyses linguistic conjunction in the following terms. By conjoining two utterances the speaker indicates that the hearer should process them together and that doing so will achieve contextual effects over and above what each conjunct individually would achieve. Examples of the sort Carston discusses are given below.

(5) John brushed his teeth and went to bed.

(6) John fell over and hurt himself.

In (5) there is a temporal sequence; in (6) there is a causal connection. Carston analyses this as inferred meaning: it is not directly encoded, but the process of enrichment may be linguistically triggered. She notes that these effects may be derived from simple juxtaposition, but by using conjunction, the speaker guarantees that there is some connection between the parts.

[7] See Sperber and Wilson (1986/95), Blakemore (1996).

I believe that Carston's approach is the right way to proceed in the case of <LINK>. The trees have to be processed together, and some connection will be inferred. The significance of such an approach is the way that pragmatics enriches the meaning between parts of syntactic structures conventionally analysed as unitary. Such is the advantage offered by mapping a 'sentence' onto a number of distinct (propositional) trees.

The other factor to be considered here is the effect of the processing domain on the way that the L-tree contributes to the overall interpretation of the utterance. The significance of the processing tasks is in terms of the order in which premises are processed and therefore the inferential effects derived. Combining these two ideas should explain the sort of interpretive effects derived from a main tree in conjunction with an L-tree.

When within the e-task, the L-tree contributes to the result of that e-task. When within the t-task, the L-tree contributes to the result of that t-task. It is within this context that the processing of the L-tree, and the consequent derivation of interpretive effects, is done.

Consider the following examples:

(7) the man angry at the decision wrote to the press.

(8) the man wrote to the press angry at the decision.

(9) angry at the decision, the man wrote to the press.

In (7), a p-predicate providing continuous input builds a restrictive L-tree within the e-task. The predicates 'man' and 'angry' are both predicated of the metavariable. Here the way they are processed together is dictated by the local goal - that is, they are both used to identify the referent within the e-task and no further effects are derived.

In (8), the L-tree is formed from discontinuous input and is non-restrictive. Here two type (p) propositions are processed together, as part of a common t-task. Thus, they occur simultaneously, and inferences can be made on the basis of this. Note that the L-tree is the minor premise, though; hence, the tendency towards an adverbial reading.

(9) gives an example of a p-predicate used parenthetically. This builds a non-restrictive, non-integrated L-tree, which forms a distinct proposition outside the t-task of the main tree. These are two separate assertions, but some connection has to be inferred between them by dint of the <LINK> relation; hence the inference that the man wrote to the press because he was angry at the decision.

In all of these cases, I assume that the process of deriving interpretation proceeds according to Relevance criteria , requiring that adequate contextual effects are achieved for no unreasonable effort (Sperber and Wilson(1986/95)). Using the <LINK> operation the speaker guides the hearer to derive extra effects; these depend on the conceptual content of the propositions, the way the L-tree is derived, the particular type of L-tree and the general (cognitive) context.

The above sketch is intended to demonstrate the potential for deriving interpretation by combining the structures built in this operation, the dynamics of the processing and the insights of Relevance Theory. A full account of these phenomena must await further research, but the promise of this line of inquiry should be clear.

I hope that this work contributes to the development of LDSNL by illuminating basic properties of the structure building system, demonstrating how the <LINK> analysis can be developed and extended, and opening up new fields of inquiry. More generally, I hope to have shown the advantages of the dynamic approach to building structure for utterance interpretation.

Bibliography

Annear Thompson, S. (1971). 'The Deep Structure of Relative Clauses'. In C. Fillmore and D. Langendoen (eds.).

Annear Thompson, S. (1973). 'On the Applicability of Root Transformations'. *Linguistic Inquiry* **4**: 465-498.

Ariel, M. (1990). *Accessing NP Antecedents*. London: Routledge.

Ariel, M. (1994). 'Interpreting Anaphoric Expressions: a Cognitive versus a Pragmatic Approach'. *Journal of Linguistics* **30**: 3-42.

Bach, E. and B. Partee. (1980). 'Anaphora and Semantic Structure', in J. Kreiman and A. Ojeda (eds.). *Papers from the Parasession on Pronouns and Anaphora*. Chicago: Chicago Linguistics Society. 1-28

Baltin, M. (1981). 'Strict Bounding', in C. Baker and J. McCarthy (eds.). *The Logical Problem of Language Acquistion*. Cambridge, Mass.: MIT Press.

Baltin, M. (1983). 'Extraposition: Bounding versus Government Binding'. *Linguistic Inquiry* **14**: 155-162.

Baltin, M. (1984). 'Extraposition Rules and Discontinuous Constitutents'. *Linguistic Inquiry* **15**:157-163.

Baltin, M. and A. Kroch. (eds.). (1989). *Alternative Conceptions of Phrase Structure*. Chicago: University of Chicago Press.

Bar-Hillel, Y. (1954). 'Indexical Expressions'. *Mind* **63**:359-79.

Beerman, D., D. LeBlanc and H. van Riemsdijk. (eds.). (1997) *Rightward Movement*. Amsterdam: John Benjamins.

Blackburn, S. & W. Meyer Viol. (1994). 'Linguistics, Logic and Finite Trees'. *Bulletin of Interest Group in Pure and Applied Logics* **1, No. 1**: 3-29.

Blakemore, D. (1996). 'Are Apposition Markers Discourse Markers?'. *Journal of Linguistics* **32**: 325-347.

Burton-Roberts, N. (1994). 'Ambiguity, Sentence and Utterance: a Representational Approach'. *Transactions of the Philological Society* **92 (2)**: 179-212.

Burton-Roberts, N. (1997). 'Language, Linear Precedence and Parentheticals'. ms, University of Newcastle upon Tyne.

Carston, R. (1988). 'Implicature, Explicature and Truth Theoretic Semantics', in R. Kempson (ed.).

Carston, R. (1993). 'Conjunction, Explanation and Relevance'. *Lingua* **90**: 27-48.

Carston, R. (1998). *Pragmatics and the Explicit/Implicit Distinction*. PhD dissertation, UCL, University of London. (Forthcoming, Oxford: Blackwell).

Chomsky, N. (1973). 'Conditions on Transformations', in S. Anderson and P. Kiparsky, (eds.). *A Festschrift for Morris Halle*. New York: Holt, Reinhart and Winston.

Chomsky, N. (1986). *Knowledge of Language: its Nature, Origin and Use*. New York: Praeger.

Chomsky, N. (1995). *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Cinque, G. (1997). *Adverbs and Functional Heads - a Cross-Linguistic Perspective*. Oxford: OUP.

Culicover, P. and L. McNally (eds.). (1998). *Syntax and Semantics*, Vol **29**: *The Limits of Syntax*. San Diego: Academic Press.

Culicover, P. and M. Rochemont. (1983). 'Stress and Focus in English'. *Language* **59**: 123-165.

Culicover, P. and M. Rochemont. (1990). 'Extraposition and the Complement Principle'. *Linguistic Inquiry* **21**: 23-47.

Davidson, D. and G. Harman (eds.). (1972). *Semantics of Natural Languages*. Dordrecht: Reidel.

Doherty, C. (1996). 'Clausal Structure and the Modern Irish Copula'. *Natural Language and Linguistic Theory* **14**: 1-46.

Donnellan, K. (1966). 'Reference and Definite Descriptions'. *The Philosophical Review* **75**, 281-304.

Doron, E. (1983) *Verbless Predicates in Hebrew*. PhD dissertation, University of Texas at Austin.

Doron, E. (1986) 'The Pronominal "Copula" as Agreement Clitic', in H. Borer, (ed.). *Syntax and Semantics,* Vol **19**: *the Syntax of Pronominal Clitics*. Orlando: Academic Press.

Emonds, J.E. (1976). *A Transformational Approach to English Syntax*. New York: Academic Press.

Emonds, J.E. (1979). 'Appositive Relatives Have No Properties'. *Linguistic Inquiry* **10**: 211-243.

Enç, M. (1991). 'The Semantics of Specificity'. *Linguistic Inquiry* **22**: 1-25.

Erteshik-Shir, N. (1973) *On the Nature of Island Constraints*. PhD dissertation, MIT.

Erteshik-Shir, N. and S. Lappin. (1979). 'Dominance and the Functional Explanation of Island Phenomena'. *Theoretical Linguistics* **6**: 41-85.

Espinal, M. T. (1991). 'The Representation of Disjunct Constituents'. *Language* **67**: 726-762.

Fabb, N. (1990). 'The Difference between English Restrictive and Nonrestrictive Relative Clauses'. *Journal of Linguistics* **26**: 57-78.

Fillmore, C. and D. Langendoen (eds.). (1971). *Studies in Linguistic Semantics*. New York: Holt, Rinehart and Winston.

Firbas, J. (1957). 'Some Thoughts on the Function of Word Order in Old English and Modern English'. *Studia Minora Facultatis Philosophicae Universitatis Brunenensis* **A5**: 72-100. Brno: Masaryk University.

Firbas, J. (1992). *Functional Sentence Perspective in Written and Spoken Communication*. Cambridge: CUP.

Fodor, J.A. (1972). 'Troubles about Actions', in D. Davidson and G. Harman (eds.).

Fodor, J. (1978). 'Parsing Strategies and Constraints on Transformations', *Linguistic Inquiry* **9**: 427-473.

Gabbay, D. (1996). *Labelled Deductive Systems*. Oxford: Blackwell.

Gabbay, D. and R. Kempson. (1992). 'Natural language Content: a Proof Theoretic Perspective'. *Proceedings of Eighth Amsterdam Semantics Colloquium*.

Gawron, J. (1985). 'A Parsimonious Semantics for Prepositions and Cause'. *Proceedings of the Regional Meeting of the Chicago Linguistic Society* **21**, Part 2, *Papers from the Parasession on Causatives and Agentivity*. Chicago: Chicago Linguistics Society, 32-47.

Goldberg, A. (1995). *Constructions: a Construction Grammar Approach to Argument Structure*. Chicago: University of Chicago Press.

Goodluck, H. and M. Rochemont. (1992). (eds.). *Island Constraints: Theory, Acquisition and Processing*. Dordrecht: Kluwer.

Green, M. (1997). *Focus and Copula Constructions in Hausa*. PhD dissertation, SOAS, University of London.

Greenbaum, S. (1969). *Studies in English Adverbial Usage*. London: Longmans, Green and Co.

Grice, H.P. (1975). 'Logic and Conversation', in P. Cole and J. Morgan (eds.), *Syntax and Semantics,* Vol **3**: *Speech Acts*. New York: Academic Press.

Grimshaw J. and E. Williams. (1993). 'Nominalisation and Predicative Prepositional Phrases', in J. Pustejovsky (ed.).

Guéron, J. (1980). 'On the Syntax and Semantics of PP Extraposition'. *Linguistic Inquiry* **11**: 637-678.

Guéron, J. and R. May. (1984). 'Extraposition and Logical Form'. *Linguistic Inquiry* **15**: 1-31.

Haegemann, L. (1988). 'Parenthetical Adverbials: the Radical Orphanage Approach', in S. Chiba (ed.), *Aspects of Modern English Linguistics*. Tokyo: Kaitakushi.

Heycock, C. and A. Kroch. (1997). 'Inversion and Equation in Copular Sentences', in A. Alexiadou (ed.). *ZAS Papers in Linguistics* **10**: 71-87.

Hilbert, D. and P. Bernays. (1939/70). *Grundlagen der Mathematik*. Berlin: Springer.

Huck, G. and A. Ojeda. (eds.) (1987). *Syntax and Semantics,* Vol **20**: *Discontinuous Constituency*. Orlando: Academic Press.

Ifantidou, E. (1994). *Evidentials and Relevance*. PhD dissertation, UCL, University of London.

Jackendoff, R. (1977). *X-bar Syntax*. Cambridge, Mass.: MIT Press.

Jackendoff, R. (1997). *The Architecture of the Language Faculty.* Cambridge, Mass.: MIT Press.

Jensen, S. (1996) *A Computational Approach to the Phonology of Connected Speech.* PhD dissertation, SOAS, University of London

Jespersen, O. (1927). *A Modern English Grammar on Historical Principles.* Phototyped Edition (1949), Copenhagen: Einar Munksgaard.

Johnson, D. and S. Lappin. (1996). A Critique of the Minimalist Program, in S. Ploch and A. Simpson, (eds.). *SOAS Working Papers in Linguistics and Phonetics* **6**: 220-273.

Kamp, H. (ms) 'SID Without Time or Questions'. ms, University of Texas at Austin.

Kamp, H. and U. Reyle. (1993). *From Discourse to Logic*. Dordrecht: Kluwer.

Karttunen, L. (1976). 'Discourse Referents', in J. McCawley (ed.), *Syntax and Semantics,* Vol **7**: *Notes from the Linguistic Underground.* New York: Academic Press

Kasher, A. (1991). *The Chomskyan Turn*. Oxford: Blackwell.

Kaye, J. (1993). 'Derivations and Interfaces', in M.Cobb and Y. Jiang (eds.). *SOAS Working Papers in Linguistics and Phonetics* **3**: 90-126.

Kempson, R. (ed.). (1988). *Mental Representations*. Cambridge: CUP.

Kempson, R. (1988). 'Grammar and Conversational Principles', in F.J. Newmeyer (ed.).

Kempson, R. and D. Gabbay. (1997). 'Ellipsis as Predicate Anaphora: Towards a Dynamic, Structural Account'. ms, SOAS, University of London.

Kempson, R. and D. Gabbay. (fthcmng). 'Crossover: a unified view'. To appear in *Journal of Linguistics*.

Kempson, R. & M. Edwards. (1998). 'Resumptive Pronouns in English and Arabic'. Talk Given at the Conference on Afro-Asiatic Languages 4, London.

Kempson, R. and W. Meyer Viol. (1998). 'Topic and Focus Structures: the Dynamics of Tree Growth' in R. Oehrle et al. (eds.). *Proceedings of the Conference on Formal, Categorial and Head-driven Grammar*, ESSLLI 1998, Saarbruecken.

Kempson, R., W. Meyer Viol, and D. Gabbay. (in prep.). *Dynamic Syntax: the Deductive Flow of Natural Language*. Oxford: Blackwell.

Kroch, A. and A. Joshi. (1987). 'Analysing Extraposition in a Tree Adjoining Grammar', in A. Huck and A. Ojeda, (eds.).

Lappin, S. (1991). 'Concepts of Logical Form in Linguistics and Philosophy', in A. Kasher, (ed.).

Lappin, S. and Francez, N. (1994). 'E-type Pronouns, I-Sums and Donkey Anaphora'. *Linguistics and Philosophy* **17**: 391-428.

Legendre, G. (1997). 'Secondary Predication and Functional Projection in French'. *Natural Language and Linguistic Theory* **15**: 43-87.

McCawley, J. (1982). 'Parentheticals and Discontinuous Structure'. *Linguistic Inquiry* **13**: 91-106.

McCawley, J. (1987). 'Some Additional Evidence for Discontinuity', in A. Huck and A. Ojeda, (eds.).

McCawley, J. (1989). 'Individuation in and of Syntactic Structures', in M. Baltin and A. Kroch (eds.).

McConnell-Ginet, S. (1982). 'Adverbs and Logical Form'. *Language* **58**: 144-184.

McNulty, E. (1988). *The Syntax of Adjunct Predicates.* PhD dissertation, University of Connecticut, Storrs.

Marten, L. (in prep.). *Syntactic and Semantic Underspecification in the Verb Phrase*. PhD dissertation, SOAS, University of London.

May, R. (1977). *The Grammar of Quantification*. PhD dissertation, MIT.

May, R. (1985). *Logical Form: its Structure and Derivation*. Cambridge, Mass.: MIT Press.

Meyer Viol, W., R. Kempson, and D. Gabbay. (1997). 'Language Understanding: a Procedural Perspective'. *Logical Aspects of Computational Linguistics, First International Conference, LACL 96. Lecture Notes in Computer Science/Artificial Intelligence* **1328**. Berlin: Springer Verlag.

Milsark, G. (1977). 'Towards an Exploration of Certain Peculiarities of the Existential Construction in English'. *Linguistic Analysis* **3**: 1-29.

Montague, R. (1974). *Formal Philosophy: Selected Papers of Richard Montague*. Edited and with an Introduction by R. H. Thomason. New Haven: Yale University Press.

Newmeyer, F.J. (ed.). (1988). *Linguistics: the Cambridge Survey II, Linguistic Theory: Extensions and Implications*. Cambridge: CUP.

Partee, B. (1972). 'Opacity, Coreference and Pronouns'. In D. Davidson and G. Harman (eds.).

Partee, B. (1987). 'Noun Phrase Interpretation and Type Shifitng Principles', in J. Groenendijk et al. (eds.). *Studies in Discourse Representation Theory and the Theory of Generalised Quantifiers*. Dordrecht: Foris.

Perrett, D. (in prep.) *The Dynamics of Tense Construal in Hadiyya.* PhD dissertation, SOAS, University of London.

Pesetsky, D. (1982). *Paths and Categories*. PhD dissertation, MIT.

Prince, E. (1998). 'On the Limits of Syntax, with Reference to Left-Dislocation and Topicalization', in P. Culicover and L. McNally (eds.).

Pustejovsky, J. (ed.). (1993). *Semantics and the Lexicon*. Dordrecht: Kluwer.

Rapoport, T. (1987). *Copular, Nominal and Small Clauses: a Study of Israeli Hebrew*. PhD dissertation, MIT.

Rapoport, T. (1991). 'Adjunct Predicate Licensing and D-Structure', in S. Rothstein, (ed.).

Rapoport, T. (1993). 'Verbs in Depictives and Resultatives', in J. Pustejovksy (ed.).

Roberts, I. (1988). 'Predicative APs'. *Linguistic Inquiry* **19**: 703-710.

Rochemont, M. (1978). *A Theory of Stylistic Rules in English*. PhD dissertation, University of Massachusetts, Amherst.

Rochemont, M. (1986). *Focus in Generative Grammar*. Amsterdam: Benjamins.

Rochemont, M. (1992). 'Bounding Rightward $\bar{A}$-Dependencies, in H. Goodluck and M. Rochemont (eds.).

Rochemont, M. and P. Culicover. (1990). *English Focus Constructions and the Theory of Grammar*. Cambridge: CUP.

Ross, J.R. (1967). *Constraints on Variables in Syntax*. PhD dissertation, MIT.

Ross, J.R. (1973). 'Slifting', in M. Gross, M. Halle, and M. Schuetzenberger, (eds.) *The Formal Analysis of Natural Language*. The Hague: Mouton.

Rothstein, S. (1983). *The Syntactic Forms of Predication.* PhD dissertation, MIT.

Rothstein, S. (ed.). (1991). *Syntax and Semantics*, Vol **25**: *Perspectives on Phrase Structure, Heads and Licensing.* San Diego: Academic Press.

Rouchota, V. (1996). 'Discourse Connectives: What Do They Link?', in P. Backley and J. Harris (eds.). *UCL Workings Papers in Linguistics* **8**.

Safir, K. (1986). 'Relative Clauses in a Theory of Binding and Levels'. *Linguistic Inquiry* **17**: 663-689.

Sag, I. (1997). 'English Relative Clause Constructions'. *Journal of Linguistics* **33**: 431-483.

Sperber, D. & D. Wilson (1986/95) *Relevance: Communication and Cognition.* Oxford: Blackwell.

Stowell, T. (1989). 'Subjects, Specifiers and X-bar Theory', in M. Baltin and A. Kroch (eds.).

Swinburne, D. (1998). '"Hu" and the Dynamics of Hebrew'. Talk Given at the Conference on Afro-Asiatic Languages 4, London.

van Valin, R. and R. la Polla. (1997). *Syntax: Structure, Meaning and Function.* Cambridge: CUP.

Williams, E. (1994). *Thematic Structure in Syntax.* Cambridge, Mass.: MIT Press.

Wilson, D. and D. Sperber. (1993). 'Linguistic Form and Relevance'. *Lingua* **90**: 5-25.

Wittenburg, K. 'Extraposition from NP as Anaphora', in G. Huck and A. Ojeda (eds.).

Ziv, Y. (1975). 'On the Relevance of Content to the Form-Function Correlation (an Examination of Extraposed Relative Clauses)', in R. Grossman et al. (eds.). *Papers from the Parasession on Functionalism.* Chicago: Chicago Linguistics Society, 566-579.

Ziv, Y and P. Cole. (1974). 'Relative Extraposition and the Scope of Definite Description in Hebrew and English', in M. LaGaly et al. (eds.). *Papers from the Tenth Regional Meeting of the Chicago Linguistics Society.* Chicago: Chicago Linguistics Society, 772-786.