Modelling an incremental theory of Lexical Functional Grammar

Stephen M. Jones

Kellogg College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2019

For my parents.

Abstract

This thesis presents and tests an incremental theory of Lexical Functional Grammar (LFG) in an attempt to support researchers in formal grammar to engage with questions raised by experimental findings on language processing and artificial intelligence. Previous work on the incremental building of syntactic structure has concentrated on constituent structure, considering interactions between word class and phrase-structure rules. However, syntax in LFG is represented not only as constituent structure, but also as functional structure: a universal set of grammatical functions is included within the primitives of the theory.

The incremental theory presented here explores the role of grammatical functions as well as category and phrase-structure rules, to build representations of c-structure and f-structure. Universal and language-specific well-formedness constraints interact with lexical content to shape the process of incremental structural growth. The theory is then used to derive predictions about the impact of context on processing decisions.

The theory is tested computationally using a model built in the ACT-R computational cognitive architecture. The model's production set assesses the combinatorial constraints that apply to lexical input in the context of previously processed material and builds structure accordingly. The combinatorial constraints are contained in lexical specifications stored in declarative, rather than procedural memory. As a result of this the model can use a single production set to process lexical input in English and Korean. The outputs of simulations show that the model is capable of building monoclausal and complex sentences in the two language. The model also simulates context-dependent variations in the parsing of identical strings. A language-independent proposal of the effect of prosodic breaks is included in the model, and this is used to simulate prosodic facilitation of the processing of complex syntax.

Acknowledgements

Throughout my DPhil project I have benefited greatly from contact with a number of people and organisations.

I would like to thank my supervisors for their complementary inputs to the project. Jieun Kiaer provided support for my understanding of Korean and incremental grammar theories, and practical advice and direction about where I should seek to present my work and gain wider feedback. Mary Dalrymple read numerous drafts of texts ranging from partially-formed ideas to whole chapters, always giving me rapid, insightful, and constructive feedback. In our discussions, she balanced encouragement, probing questions, and practical advice, making sure that I had the freedom to shape the project and determine its direction. I learned much from her linguistic knowledge and her experience as a supervisor, and I hope one day to make a similar contribution to other people's work.

I would also like to thank my examiners, John Lowe and Shravan Vasishth for their stimulating questions and their encouragement to continue with my work after the DPhil.

I was fortunate to be part of the Faculty of Linguistics, Philology, and Phonetics in Oxford, where I started learning about linguistic theory, and where I enjoyed many productive conversations. Alongside my tutors, I am particularly grateful for time spent with Jamie Findlay, Yoolim Kim, Joey Lovestrand, Louise Mycock, and other friends and colleagues not named here. Yoolim Kim and Myungsu Kang patiently answered my numerous questions on Korean example sentences. I tested ideas with, and received valuable feedback from, the faculty Syntax Working Group. I also benefited from the feedback given by my examiners for transfer and confirmation of status: Ash Asudeh, Aditi Lahiri, and Matt Husband, who suggested that I explore cognitive models of language.

The cognitive modelling element of the thesis was supported by many people at the University of Groningen. The tutors and other participants at the 2017 and 2018 Cognitive Modelling Spring Schools enabled my development of skills in ACT-R and Lisp coding. I am indebted to Jacolien van Rij-Tange, who acted as informal supervisor during my study visit to Groningen, and to Niels Taatgen, who discussed the limitations of ACT-R and provided advice on technical questions.

I was able to present my work at a number of conferences and workshops: Dynamic Syntax 2017 and 2018, ICCM/MathPsych 2017, LAGB 2017, LFG 2017 and 2018, and the regular SE-LFG workshops. My thanks go to the organisers and other participants for the learning that took place during those presentations and subsequent discussions.

The process of actually generating text was made much easier by the Oxford Humanities Academic Writing Group and Sainsbury's *Taste the Difference*™ Assam tea bags.

And most of all I thank my husband David, without whose love, patience, and unstinting encouragement this thesis would not have been completed.

List of abbreviations

A	adjective/adverb
ACC	accusative
Adj/ADJCV	adjective
ADJ	adjunct
ADM	Argument Dependency Model
ADN	adnominal
Adv/ADV	adverb
ARG	argument
BEN	benefactive
C	complementiser
CAT	category
CCG	Combinatory Categorial Grammar
COMP	complement
COORD	coordination
CP	complementiser phrase
DAT	dative
DEF	definite
DF	discourse function
D	determiner
DP	determiner phrase
eADM	extended Argument Dependency Model
ERP	event-related potential
fMRI	functional magnetic resonance imaging
FUT	future
GEND	gender
GEN	genitive
GF/GF	grammatical function
INDIC	indicative
INF	infinitive
LDD	long-distance dependency
LFG	Lexical Functional Grammar
LHS	left hand side
LOC	locative
M	masculine
MRC	mental representation of what is to be communicated
N	noun
NP	noun phrase
NOM	nominative
NUM	number
OBJ	object
OBL	oblique
ORC	object relative clause
P	preposition
PCASE	prepositional case
PERS	person
PET	positron emission tomography
POL	polite
POSS	possessive
PP	prepositional phrase
PRED	predicate
PRET	preterite
PRON	pronoun (word class)
PRO	pronoun referent
PRS	present
PST	past
RELPRO	relative pronoun
RC	relative clause
REL	relative
RHS	right hand side
RRC	restricted relative clause
S	clause
SBJ	subject
SG	singular
SRC	subject relative clause
SUBJ	subject
TAM	tense/aspect/mood
TOP	topic
V	verb
VCOMP	restricted verb phrase complement
VP	verb phrase
WH	content question/ relative pronoun
XADJ	open clausal adjunct
XCOMP	open clausal complement

Contents

Chapter 1

Introduction

> *"Perhaps the closest partnership in cognitive science is the one between psycholinguistics and formal linguistics. The two disciplines were born together and have grown up essentially side-by-side as sister disciplines. But like many family relationships, the dynamics between these fields have been complicated."*
> (Ferreira, 2005, p. 365)

1.1 Background

The above quotation is taken from a 2005 programmatic paper by psycholinguist Fernanda Ferreira in which she raises concerns about the lack of contact between the research agendas of psycholinguistics and formal linguistics, and sets out recommendations to improve collaboration between the two areas of linguistics. When addressing generative grammar, Ferreira was referring to theories addressing competence-based linguistic knowledge (Chomsky, 1965) and specifically discussed Government and Binding theory (Chomsky, 1981) and *The Minimalist Program* (Chomsky, 1995).

Ferreira was writing ten years after the publication of *The Minimalist Program* and her frustration with the increased level of abstraction away from the surface string is evident: she describes the Minimalist Program as "highly unappealing from the point of view of human sentence processing." (Ferreira, 2005, p. 370). The paper closes with three challenges to formal linguists. First, they are challenged to pay more attention to the quality of evidence used to support theoretical claims. Second, they are challenged to pay more attention to developments in cognitive science beyond psycholinguistics. Third, given the evidence from psycholinguistics that syntax, semantics and phonology are all closely associated with sentence processing, they are asked to consider how formal treatments of these three aspects of language of might be reintegrated, rather than focus on syntax to the exclusion of other sub-disciplines within linguistics.

Although Chomskyan generative grammar remains in 2019 the predominant theoretical approach to syntax, it is far from the only formal theory of grammar. Other established theories assume transparent connections between grammatical constraints and the surface form of language, with syntax, semantics and phonology operating as interdependent, mutually constraining elements of the grammar.

2005 also saw the publication of an influential computational model of sentence processing (Lewis and Vasishth, 2005), which used the general principles of cognition within ACT-R, a computational cognitive environment (Anderson, 1990, 2007) to show that human performance on the processing of complex syntax could be modelled assuming a cue-based retrieval process for building syntactic structure. Fourteen years after publication, this model remains a benchmark for other modelling approaches, and has been extended both in the range of phenomena modelled and the languages for which modelling data have been produced.

Matching human performance in language processing is not restricted to psycholinguistics or to time-course data. Since 2005 there has been a significant expansion in the fields of machine learning and artificial intelligence. The goal of computers responding in human-like ways to natural language input is long-standing, and models that incorporate grammatical knowledge within computational cognitive environments continue to be developed to address this goal.

1.2 Content of the thesis

This thesis sits at the interface between formal theories of grammar, psycholinguistics, and cognitive modelling, and is intended to be of interest to researchers in all three disciplines. In the thesis I employ an established formal linguistic theory, Lexical Functional Grammar, LFG (Kaplan and Bresnan, 1982). LFG is a modular, declarative, constraint-based theory that attempts to map the relationship between language form and meaning, and includes not only syntax, but also semantics, information structure, and phonology. LFG is thus well-placed to take up one of Ferreira's three challenges.

LFG is amenable to analyses of sentence fragments and Asudeh (2012) set out principles for how LFG might engage with questions of language performance. I now extend that work by elaborating an incremental theory of LFG that shows how universal and language-specific constraints operate incrementally to constrain the building of language structure, and without requiring top-down assumptions about the ultimate structural goal of parsing. I then use that grammar as the basis for a computational model of structure-building in ACT-R. Within the model, I introduce a new structure within LFG, which I term the F-representation. This does not replace the c-structure and f-structure levels of representation that are part of the established theory, but allows the model to build a single structural representation from which c-structure and f-structure representations are recoverable.

The model focuses on the initial stages of structure-building and uses a single set of production rules to process lexical input from two typologically different languages, English and Korean. I show how the model is able to generate similar representations from sentences with similar meanings in the two languages, despite the very different constituent structure of the sentences. The model is able to replicate human performance on ditransitive garden path sentences in English, using the constraints inherent in the incremental grammar. The model also incorporates a representation of prosodic boundaries, which are used very early in human sentence processing to support decisions about how to build structure when multiple options are available. I demonstrate how prosodic boundaries can facilitate or inhibit the processing of complex sentences, depending on their placement, again matching experimental data on human performance.

1.3 Structure of the thesis

The thesis is structured as follows. In Chapter 2, I review the literature on questions that are important in the development of incremental theory. These include experimental studies of the role of grammar in processing and accounts of the cognitive organisation of sentence processing, which underpin my assumed processing pathway and support the delimitation of the scope of the cognitive model so that it can focus on phenomena specific to grammar and interactions with prosody. I consider evidence on the prosody-syntax interface, which supports my assumptions in the model about the function of prosodic breaks. I also critically review existing incrementally-developed hierarchical representations of language structure, to motivate the development of an incremental theory of LFG.

In Chapter 3, I give a brief introduction to LFG, and then elaborate an incremental theory that incorporates universal and language-specific constraints on the growth of constituent and functional structures. I identify specific predictions about how these constraints will interact during processing, and show how the incremental theory analyses language input from Korean, using the grammar provided by Cho and Sells (1995). This worked example is then used to discuss the lexical, syntactic, prosodic, semantic and discourse factors that come into play at points during processing where structure-building is possible at more than one site.

Chapter 4 introduces cognitive modelling and cognitive architectures as tools for describing and testing hypotheses about language processing. I provide an overview of the structure and function of ACT-R, and critically review the role of grammar in models of sentence processing in ACT-R and other cognitive architectures. I also identify the architectural assumptions and other issues to take into account when designing a cognitive model, including the desirability of a single structural representation.

In Chapter 5, I place the incremental theory of LFG within an assumed pathway of sentence processing, and specify the limits of the modelling task. I then apply the findings from the review of language processing models to the incremental theory of LFG. As a consequence of this I introduce the F-representation as a single representation of syntactic structure for modelling purposes, whose content is formally derivable from c- and f-structure. I also consider assumptions about the architecture of the model, including the use of additional ACT-R buffer capacity, and how working memory is created and managed during processing.

Chapter 6 describes the structure and operation of the ACT-R model that I have built taking into account the issues discussed in Chapter 5. This includes discussion of memory structures, the production set that is used to build structure, and features of the English and Korean lexicons. I present the outputs of the model using English and Korean inputs for the tasks of building syntactic structure, testing predictions relating to the choice of attachment site during processing, the emergence/avoidance of garden path phenomena, and the impact of prosodic breaks. The limitations of the model and ACT-R, and the implications of the results are discussed, together with potential areas for future development of language models, and possible structural amendments to ACT-R.

Finally, in Chapter 7, I review the claims and findings presented in the thesis. Alongside the future research agenda for cognitive modelling that was presented in Chapter 6, I identify potential directions to extend the coverage of the incremental theory to include s-structure and discourse representations.

1.4 Delimiting the thesis

LFG as a declarative, constraint-based theory, is amenable to descriptions of production and comprehension: the model in this thesis is limited to comprehension. The model works with two typologically different languages, English and Korean, but both of these languages have endocentric phrase structure which contributes to the assignment of functional relationships. Languages with highly free word order are not considered. The model also excludes the construction of sets that allow multiple f-structures to refer to a single discourse referent (Sadler and Nordlinger, 2010). The aim of the model is to explore structure-building cross-linguistically. As such, model fit is assessed by testing the structures produced against native-speaker interpretation. High-level model time-course data is presented, but matching detailed time-course variations is outside the scope of the project.

Chapter 2

The role of grammar in theories of processing

It is not possible within the constraints of this thesis to review all of the literature on grammar and processing, but it is important to understand how the incremental theory of grammar set out in Chapter 3 relates to theories of processing, and also to ground the assumptions that underpin the cognitive model of the incremental theory, which is developed in Chapter 5 and presented in Chapter 6.

This chapter focuses on four areas that are most salient to the remainder of the thesis. The first area is a brief overview of the experimental evidence on the relationship between language processing and native speakers' knowledge of the syntactic elements of grammar, together with the relationship between prosody and the interpretation of language structure. The second area sets out some of the proposals about the organisation of language processing tasks within cognition: what sub-tasks have been proposed within the overall processing task; what proposals have been made as to how these sub-tasks relate to each other sequentially and/or hierarchically; and how the relationship between language processing and other elements of cognition has been characterised. The third area considers the evidence on interactions between prosody and the processing of syntactic structure. Finally, the fourth area considers theoretical representations of hierarchical language structure, addressing theories of grammar that specifically consider the inherent ordering of language as well as the totality of an utterance when accounting for grammaticality.

2.1 Phenomena

In this review of phenomena considered in psycholinguistic experiments, I am considering three areas where there is strong evidence for an effect of syntactic structure on processing: garden path effects, structurally-linked time-course variations, and modulation of time-course variations linked to the operation of higher-level syntactic constraints.

2.1.1 Structural ambiguity during processing

Structural ambiguity in language is pervasive, and does not necessarily cause a processing difficulty that is perceived by readers or hearers. For example, MacDonald et al. (1994) illustrate provide examples of temporary structural ambiguities taken from the literature on sentence processing, which they claim appear on average around once per journal page.

Frazier and Clifton (1996, pp. 11–12) give a typology of temporarily ambiguous sentence structures including those with a perceptible garden path effect. Apart from (3a) the examples are all from English. Although all of the examples they give result in structural ambiguity, the characteristics of the words and contexts in their examples are varied. They include lexical ambiguity between finite and non-finite verb forms (1a), verbs with valency alternations in different clausal contexts (1b)–(1d), shifts in the scope of conjunction (1e), multiple attachment sites for dependents (2) including ambiguities enabled by valency alternations (2e), and structures where alternative readings are available either because of ambiguity in the head of a noun compound, alternation in grammatical function available because of pro-drop, or semantic scope (3).

Frazier (1978) investigated the time-course of processing sentences with ambiguous attachments or garden path effects and used this as evidence that two principles guide syntactic structure building. These were *Late Closure*: "When possible, attach incoming material into the clause or phrase currently being parsed," and *Miminal Attachment*: "Attach incoming material into the phrase marker being constructed using the fewest nodes consistent with the well-formedness of the language." (*ibid.*, p. 111). I return to Frazier's work in Section 2.2.

(1) Structural reanalysis of element in *italics*

a. Main clause ⟶ reduced relative

The horse *raced past the barn* fell.

b. Object of modifying clause ⟶ subject of matrix clause

While Mary was mending *the sock* fell off her lap.

c. Object of matrix clause ⟶ subject of complement clause

John knew *the answer to the physics problem* was wrong.

d. Indirect object of matrix clause ⟶ subject of reduced relative clause

Fred gave the man *the dog* bit the package.

e. NP conjunction ⟶ S conjunction

Jacob kissed *Miriam and her sister* laughed.

(2) Multiple possible attachment sites

a. PP attachment to VP or NP

I saw the girl with the telescope.

b. Attachment of PP to higher or lower clause

I put the book that you were reading *in the library.*

I put the book that you were reading *in the library* into my briefcase.

c. Attachment of S to higher or lower clause

Fred will realise that Mary left *when the party starts.*

Fred will realise that Mary left *when the party started.*

d. Attachment of Adverb to higher or lower clause

We remembered that the assignment will be due *yesterday.*

We remembered that the assignment will be due *tomorrow.*

e. Attachment of complement to VP or NP

John told the girl *that Bill liked the story.*

John told *the girl that Bill liked* the story.

(3) a. Sentential object with pro-drop subject and a right-dislocated topic

Ha chiamato Giovanni
has called Giovanni

"Someone has called Giovanni." OR "Giovanni has called."

b. Purpose clause vs rationale clause

Nixon$_i$ bought a 1960s version of Trivial Pursuit$_j$ (pro$_j$/pro$_i$) to amuse his friends.

The sentences in (1) are considered by Frazier and Clifton to produce perceptible garden path effects. I will consider these in Section 2.1.1.1. The ambiguities from the sentences in (2) are argued to have preferred readings which may be affected by context, and I will consider these in Section 2.1.1.2.

2.1.1.1 Garden path effects

The phenomenon of garden path sentences has long been taken as evidence to support the hypothesis that a single structure is built incrementally during language processing. Perhaps the classic example is (4), taken from Bever (1970).

(4) The horse raced past the barn **fell**.

A temporary illusion of ungrammaticality is experienced at the target word **fell**. Up to that point the sentence can be parsed monoclausally with *raced* as its matrix verb. However, the word *fell* can be parsed as a finite verb which requires a preceding subject in order for the sentence to be grammatical. Thus the preceding string must be reanalysed making *raced past the barn* an embedded clause modifying *the horse*, which then becomes the subject of *fell.* The verb *raced* can be parsed either as a finite verb which can head a matrix clause, or as a non-finite participle which can head an embedded clause: the effect is then accounted for by assuming that a choice must be made when a point of ambiguity is reached, and that the context at that point favours the matrix reading of *raced* over the participle.

One feature of garden path sentences is that they are cancellable by changing the context, e.g. substituting an unambiguous participle for *raced* (5a), delimiting the embedded clause with punctuation or prosody (5b), adding words that make the hierarchical structure explicit (5c), or changing the context to make a different reading of *fell* more likely thus leaving the matrix interpretation of *raced* unchallenged (5d).

(5) a. The horse *driven* past the barn fell.
    b. The horse, raced past the barn, fell.
    c. The horse that was raced past the barn **fell**.
    d. The horse raced past the *desolate* fell.

The idea that a clause boundary may be hidden and then later revealed is appealing. However, garden path sentences are not a unitary phenomenon, nor are they an absolute phenomenon. Instead they can be seen as a special case of temporary ambiguity in structural sentence processing. However, as Sturt and Crocker (1998) point out, not all unmarked clause boundaries are equal: example (6) is adapted from their paper. In each case the italicised sequence V + NP contains the boundary of an embedded clause (7), but garden path effects are perceived at the word **was** for (6a) but not in the case of (6b).

(6) a. While the guests *ate the cake* **was** still being decorated.
    b. The guests *saw the cake* **was** still being decorated.

(7) a. $[_{\mathrm{S}}$ $[_{\mathrm{CP}}$ While the guests *ate* $]_{\mathrm{CP}}$ *the cake* was still being decorated.$]_{\mathrm{S}}$
    b. $[_{\mathrm{S}}$ The guests *saw* $[_{\mathrm{CP}}$ *the cake* was still being decorated. $]_{\mathrm{CP}}$ $]_{\mathrm{S}}$

The nature and ease of repair processes has also been investigated. A sentence containing a misparsed ambiguity may be judged as grammatical once repair has taken place, even if that repair required some effort, or the repair process might be too difficult, resulting in a judgement of ungrammaticality. For example, Ferreira and Henderson (1991) identified that the longer the distance between the head of a hidden ambiguous phrase and the ambiguity being revealed, the less likely it was that a reanalysis was possible. Bailey and Ferreira (2003) showed that interrupting the fluency of speech after a hidden ambiguity also inhibited subsequent repair processes once the ambiguity was apparent.

There is evidence that event representations generated by the initial, unsuccessful parse of the first element of a garden-path sentence persist, even after the whole sentence has been parsed successfully. Thus Christianson et al. (2001) report on an experiment where subjects read sentences such as example (8) with the presence or absence of a comma serving to cancel a potential garden path as in (5b).

(8) While Anna dressed(,) the baby that was small and cute spit up on the bed.

Where no comma was present, and so the initial phrase was not clearly marked as an adjunct, 60% of the subjects answered that Anna had dressed the baby, a conclusion that cannot be drawn compositionally from the final sentence. Even where a comma was present to inhibit the garden path reading *While Anna dressed the baby...*, 12% of subjects reported a meaning inconsistent with compositional semantics. Slattery et al. (2013) investigated this finding further and conclude that the persistence of the interpretation from the incorrect reading is not due to a failed repair process, but rather to representations of earlier interpretations remaining in memory to compete with later interpretations.

2.1.1.2 Lexical influence on attachment

Factors other than syntactic structure also play a role in attachment. Even where there is a structural ambiguity arising from the valency alternation of the verb *ate* in (6a), the meaning of the following noun may either reduce the garden path effect or make reanalysis easier (Ferreira, 2005): compare (6a) with the structurally identical sentence in (9).

(9) While the guests ate the hall was still being decorated.

Also, some syntactic structures appear to be preferred over others on-line, such as in example (10) adapted from Fodor and Ferreira (1998). In this sentence pair, no garden path effect is seen at the reflexive pronoun **herself** in (10a), but an effect is triggered at **himself** in (10b). Here, there are valid grammatical analyses for both readings (11), arising from scope interactions between genitive *'s* and the object of preposition *of* but the variant (10b) is strongly dispreferred by native speakers.

(10) a. The daughter of the teacher's son enjoyed **herself** at the party.
b. ? The daughter of the teacher's son enjoyed **himself** at the party.

(11) a. The *daughter* [of the teacher's son] enjoyed *herself* at the party.
b. [The daughter of the teacher]'s *son* enjoyed *himself* at the party.

Turning to the sentences with multiple attachment possibilities in (2), these were also argued by Fodor to be subject to the Minimal Attachment and Late Closure Strategies. However, this was found not to be the case in all languages. Cuetos and Mitchell (1988) explored attachment preferences for relative clauses in English and Spanish, using examples such as (12), where in both English and Spanish the relative clause *que tuvo el accidente* "who had had the accident" could potentially modify either *coronel* 'colonel' or *hija* 'daughter'.

(12)

El	*periodista*	*entrevisto*	*a*	*la*	*hija*	*del*	*coronel*	*que*	*tuvo*	*el*	*accidente*
The	journalist	interviewed	to	the	daughter	of.the	colonel	who	have.PRET	the	accident.

"The journalist interviewed the daughter of the colonel who had had the accident."

Cuetos and Mitchell found that in English speakers' preferred interpretation of the sentence, the relative clause modified *colonel*. This indicated use of the Late Closure strategy and replicated Fodor's findings. However, Spanish speakers preferred an interpretation of the sentence where the relative clause modified *hija* 'daughter'. Time-course experiments showed that the judgements in Spanish were made within a similar timescale to those for English, excluding the possibility that a semantic reanalysis was taking place. The authors concluded that Fodor's parsing strategies were language-specific rather than universal.

Ford et al. (1982) investigated the role of lexical variation on clausal attachment preference of prepositional phrases in English — ambiguity type (2b) from Frazier and Clifton's typology. For sentences such as those in (13), experimental participants were asked to choose between two interpretations (14), one of which was consistent with the PP attaching low to the object NP (14a), and the other of which was consistent with a benefactive reading, where the PP attaches high to the VP (14b).

(13) a. Joe *included* the package **for Susan**.
b. Joe *carried* the package **for Susan**.

(14) a. The package for Susan was included by Joe.
b. It was for Susan that Joe included the package.

The authors found substantial variation between preferences depending on the verb chosen. Thus for *included* 65% of participants preferred the low attachment reading, whereas for *carried* 90% of participants preferred the benefactive reading, in contravention to Frazier's Late Closure strategy. The authors concluded that syntactic closure was modulated by lexical information, and that lexical memory for verbs included knowledge of the probabilities of different valency alternations, which in turn affected attachment preferences on-line.

2.1.1.3 Structural ambiguity: summary

The resolution of structural ambiguities, and garden path phenomena, can be seen as evidence that hierarchical structure is built without delay as language is processed, and that building this structure requires rapid choices to be made. The presence or absence of a lexical ambiguity can be modulated by sentence context and lexical content, including punctuation in written text. Frazier's Minimal Attachment and Late Closure strategies have success in accounting for the presence of garden path phenomena in English, but the attachment preferences for modifiers show variation between languages, and within a language according to lexical content.

Once an attachment site has been chosen, that choice can be amended in the light of subsequent language content, and there is variation in the ease of the repair process. This variation is driven by factors including the distance from the repair site to the site at which the need for repair becomes apparent, and the degree of plausibility in real world terms of the initially constructed meaning. It also appears that the concept of "repair" does not necessarily entail the removal or suppression of a syntactic structure previously created, but may instead involve the modification of another structural representation that is constructed after the initial syntactic representation has been built.

2.1.2 Time course effects

The Derivational Theory of Complexity proposed by Fodor and Garrett (1967) put forward the fundamental hypothesis that increased complexity in a sentence will correlate with processing difficulty, and that increased processing difficulty is reflected in increased processing time. Although the measure of complexity proposed in Fodor and Garrett's model — the number of transformations required to derive the surface structure of a sentence from its base structure according to the generative grammar of Chomsky (1965) — was not successful in modelling the time course of processing, the association between structural complexity and processing difficulty is a common linking hypothesis in psycholinguistics.

Relative clauses introduce complexity not only by embedding a clause, but also because a long-distance dependency exists between the relative pronoun, if present, and the verbal head of the relative clause. This is associated with longer processing times for relative clauses compared to matrix clauses. Examples of a subject relative clause (SRC) and an object relative clause (ORC) are given in italics in (15): the position that the relative pronoun would occupy in the relative clause if that clause was an independent matrix clause is indicated thus: $\_\_\_\__i$.

(15) a. SRC: The man *who*$_i$ $\_\_\_\__i$ *chased the dog* argued with its owner.
b. ORC: The man *who*$_i$ *the dog chased* $\_\_\_\__i$ argued with its owner.

Many researchers have reported an asymmetry between processing times for subject relative clauses (SRC) and object relative clauses (ORC), and this asymmetry has been reported cross-linguistically regardless of the basic word order of a clause, or the morphological signals present about the nature of the clause (e.g. Baudiffier et al., 2011; Betancort et al., 2009; King and Just, 1991; Holmes and O'Regan, 1981; Kwon et al., 2010; Mak et al., 2002; Miyamoto and Nakamura, 2003; Traxler et al., 2002). This led to proposals such as the Dependency Locality Theory of Gibson (2000) that linked a specific structural analysis of the phenomenon to the time course. However, these asymmetries are modulated by lexical semantic factors such as animacy and plausibility, which suggests that grammatical function interacts with other aspects of processing. There is also evidence that the asymmetry is not necessarily universal. Gibson and Wu (2013) among others report an advantage for object relative clauses in Chinese and Kwon et al. (2013), comparing Chinese with Korean, suggests that morphosyntactic factors may also play a role in Chinese. Thus there is a demonstrated impact of embedding, but it is not clear that the asymmetry is due to syntactic rather than other linguistic factors.

Stowe (1986, 1992) compared the time course of processing long distance dependencies (LDD) that were resolved at different points in otherwise similar relative clauses. She identified that processing slowed at each point in the sentence where the long distance dependency could be resolved, shown in (16).

(16) a. My brother wanted to know if Ruth will bring us home to Mom at Christmas.
b. My brother wanted to know who$_i$ _____$_i$ will bring *us* home to *Mom* at Christmas.
c. My brother wanted to know who Ruth will bring _____$_i$ home to *Mom* at Christmas.
d. My brother wanted to know who Ruth will bring *us* home to _____$_i$ at Christmas.

She found that compared to the sentence containing an embedded clause with no LDD (16a), there were delays at the resolution sites _____$_i$ as predicted, but no delays at the argument position sites after the LDD had been resolved. However, in sentence (16d), a slowdown was also seen at the argument site '**us**' before the LDD was resolved. This is seen as evidence that when an LDD is unresolved, there are active attempts to resolve it which add to processing time, even when subsequent input shows that resolution at that point was not possible.

The processing time for resolving LDDs can be further modulated by non-structural factors such as the plausibility of the referent noun at the point of resolution. Traxler and Pickering (1996) explored this using verbs with valency alternations, which allow for an LDD to be resolved either as the direct object of the verb or as the object of a subsequent preposition. Thus in example (17), *book* is plausible both as the object of *wrote* in position 1, and also as the object of *wrote about* in position 2, whereas in (18), *city* is a plausible object of *wrote about* in position 2, but an implausible object of *wrote* in position 1.

(17) We like the book$_i$ that$_i$ the author wrote 1_i unceasingly and with great dedication about 2_i while waiting for a contract.

(18) We like the city that the author wrote $^?1_i$ unceasingly and with great dedication about 2_i while waiting for a contract.

Eye tracking fixation times were used as a measure of processing speed. When processing sentence (17), the processing delay was longer at position 2 than at position 1. This suggests that the LDD had been resolved immediately after the verb *wrote*, with a repair process being triggered when no object was encountered after the preposition at position 2. Conversely in (18), where *city* is an implausible object of *wrote*, the processing delay at position 1 was longer than that at position 2. Between the two sentences, (17) had a shorter delay than (18) at position 1, but a longer delay at position 2. The authors conclude that in sentence (18) the initial resolution of the LDD was cancelled following semantic assessment of the verb-object relationship, meaning that there was no additional computation required at position 2.

Besides variations in processing speed, researchers have also investigated phenomena where measures of brain activity are correlated with particular types of processing task, after a time delay. Two such phenomena are time-linked event-related potentials (ERP), electrical signals of brain activity measured by electroencephalogram (EEG): the N400 effect, a negative potential occurring with a latency of around 400ms (Kutas and Hillyard, 1980) after a stimulus is perceived; and the P600 effect, a positive potential with a latency of around 600ms (Neville et al., 1991). The N400 occurs around 400ms after a word is encountered and, in broad terms, the size of the effect is linked to the semantic likelihood of the word in its context (Kutas and Hillyard, 1984). The P600 seems to be linked, again in broad terms, to syntactic processing including violation of syntactic constraints and garden-path effects requiring repair (Gouvea et al., 2010). The characterisation of the N400 and P600 as "syntactic" and "semantic" respectively belies a finer-grained description. For example, the size of the P600 effect can be modulated by semantic manipulation of thematic roles (e.g. Kos et al., 2010). The timing of the effects can also vary, with effects that could be characterised as an N400 appearing as early as 200ms after the stimulus, discussed in Kutas and Federmeier (2011).

Thus the nature of language input has an impact on the time-course of processing, whether this is because it is unexpected, its structure requires additional computation (e.g. LDDs), or the initial structure calculated is defective and requires repair (e.g. garden paths). Besides these direct effects of language impact and grammar, it is also possible to explore how grammatical knowledge of syntactic structure and structural constraints affects processing, by examining how the effects discussed above, and other processing time-course phenomena, appear in different structural contexts.

2.1.3 Effects of structural constraints during processing

Resolution of anaphors adds time to processing, and Kazanina et al. (2007) explored whether the grammatical constraint on the co-reference of a pronoun in one clause with a noun in an embedded clause has an impact on processing time courses. The authors use Principle C of the Binding Theory of Chomsky (1981) to describe the constraint. Thus in sentence (19) the pronoun *she* may co-refer with *Erica*, which is in a clause that dominates it, but not *Kathryn*, which is dominated by it.

(19) Because last semester she$_{*i/j}$ was taking classes full-time while Kathryn$_i$ was working two jobs to pay the bills, Erica$_j$ felt guilty

The hypothesis, supported by data from van Gompel and Liversedge (2003), is that on encountering the pronoun *she*, a search for its antecedent is triggered, similar to the process of resolving an LDD described in by Stowe. If the grammatical constraint on co-reference has an impact on processing, there should be no processing delay found at *Kathryn*, even though in linear terms it is the first human noun encountered after the pronoun. This prediction was matched by the data, and so the authors conclude that the constraint is actively calculated during processing.

Similarly, when Traxler and Pickering (1996) repeated their experiment with the verb *wrote* inside an embedded clause and thus unavailable as an attachment site (20), they found no evidence of processing delay at position 1, nor of a difference between eye-tracking fixation times at position 2.

(20) a. We like the book$_i$ that$_i$ the author [who wrote $*1_i$ unceasingly and with great dedication] saw 2_i while waiting for a contract.

b. We like the city$_i$ that$_i$ the author [who wrote $*1_i$ unceasingly and with great dedication] saw 2_i while waiting for a contract.

Again the conclusion is that the syntactic constraint on accessibility of the LDD resolution site is actively calculated during processing.

Sturt (2003) considered the effect of gender on the processing of reflexives, looking at sentence sets similar to those in (21).

(21) a. The surgeon$_i$ who treated Jonathan$_j$ pricked himself$_{i/*j}$ with a used syringe needle.

b. The surgeon$_i$ who treated Jennifer$_j$ pricked himself$_{i/*j}$ with a used syringe needle.

c. The surgeon$_i$ who treated Jonathan$_j$ pricked herself$_{i/*j}$ with a used syringe needle.

d. The surgeon$_i$ who treated Jennifer$_j$ pricked herself$_{i/*j}$ with a used syringe needle.

In all four cases the reflexive pronoun is syntactically constrained to co-refer with *surgeon*: the author uses Principle A from Chomsky's (1981) Binding Theory to account for this. In sentences (21c)–(21d) the reflexive pronoun conflicts with the stereotypical gender of *surgeon*, although the noun has no syntactic gender and refers to both female and male surgeons. The distractor noun, Jonathan$_j$/Jennifer$_j$ matches the stereotypical gender of *surgeon* in (21a) and (21c), and matches the gender of the reflexive pronoun in (21a) and (21d). Thus different effects were predicted depending on whether or not syntactically inaccessible distractor noun did actually function as a distractor, and different patterns may be seen depending on the syntactic or the semantic computations taking place. Although later reanalysis (Jäger et al., 2019) suggests that Sturt's studies are likely to be statistically underpowered, his results suggested that the syntactic constraint on co-reference of reflexive pronouns acts early and is contradictable by other constraints (e.g. semantic plausibility). These constraints do not act before the co-reference constraint: the picture that emerges of syntactic assignment followed by, or alongside, semantic computation is similar to the findings of Traxler et al. (1998) with regard to LDD resolution.

Chow et al. (2015) explored whether syntactic structure is calculated rapidly enough such that it can be part of the context that modulates the size of an N400 effect produced by a particular word. In the sentence pair at example (22) reproduced from Chow et al. p.12, the same three NPs and matrix verb *inquired* precede the embedded target verb **evicted**.

(22) a. The exterminator *inquired* [which neighbor the landlord had **evicted** from the apartment complex].

b. The neighbor *inquired* [which exterminator the landlord had **evicted** from the apartment complex].

Although the same three NPs precede the target verb and so are part of the sentence context, the configuration of them between the matrix clause and the subordinate clause is different. In (22a), the arguments of the embedded verb **evicted** are *neighbor* and *landlord*, whereas in (22b), the arguments of the embedded verb are *exterminator* and *landlord.* This affects the cloze probability of **evicted** between the two sentences: in (22a) it has a probability of 0.22, whereas in (22b) it has a probability of 0.008. Chow et al. argue that the difference as assessed off-line by cloze probability can only modulate the N400 effect observed in processing if syntactic structure is processed rapidly enough, such that the first noun in the sentence is recognised as being in a different clause from the other two nouns by the time the embedded verb is read. However, if the syntactic structure was not calculated in time, there should be no difference in the size of the N400 effect during processing, because all three nouns are present in the prior context of the target verb in both cases.

Chow et al. report a much larger N400 ERP for (22b) than (22a) and claim that this is evidence for rapid identification of the arguments of the embedded predicate even before the predicate is heard. If the syntactic structure had no on-line effect, they argue that there should be no difference in the N400 ERP between the two sentences.

Dillon et al. (2013) proposes that different structure-building processes may show different patterns of interference with syntactic constraints, listing subject-verb agreement, processing of LDDs (e.g. wh-extraction, relative clauses), licensing of negative polarity items, ellipsis of verb phrases, and search for the referents of anaphors. He argues that this is linked to the nature of search cues that are used for the memory retrievals that form part of the cognitive process of structure building. Dillon gives experimental evidence to show that for subject-verb agreement, the number feature from a syntactically inaccessible NP can cause processing interference. However, for the binding of reflexives, syntactically inaccessible NPs do not cause interference (although other things are going on with the impact of number mismatches for grammatically accessible NPs). This suggests that cues operate differentially for different grammatical processes, although again the studies underlying Dillon's conclusions are likely to be underpowered (Jäger et al., 2019).

2.1.4 Summary of phenomena

In summary, it seems that there are some elements of processing where syntactic structure is either the sole or the principal determinant of processing, but these are limited to the early stages of processing. After that point, factors such as plausibility come into play, and syntactic constraints become weaker. This thesis is examining the role of grammatical functions as syntactic constraints on structure-building. To exclude the impact of other factors, I will therefore focus on the very earliest stages of structure building.

2.2 Accounts of the cognitive organisation of language processing

Various accounts have been proposed of the position of language processing within wider cognition, and of the configuration of sub-tasks within the overall processing task. Table 2.1, reproduced from Murphy et al. (2018, Fig. 3.1, p. 56), gives the time course for the comprehension of written or spoken language, derived from experimental measurements of brain activity during processing tasks. This is a high-level model, and the exact boundaries of each task are not specified. However, it is helpful to place the syntactic structure-building in a broader context, and I will refer to this timetable in Chapter 5 when considering how the levels of representation within LFG might map onto an assumed processing pathway.

Table 2.1: Timing of language processing tasks (Murphy et al., 2018)

Time course	Tasks per Murphy et al.	
0-100ms	Stimuli perception	
ca. 150ms	Intelligibility	
120-200ms	Local structure building	
300-500ms	Semantic analysis	*possibly in parallel*
300-500ms	Syntactic analysis	
ca. 600ms	Integration of syntax & semantics	
> 600ms	Interpretation	

In this section I will look at three questions: the extent to which the task of building a structural representation might be described as a module, the extent to which sub-tasks occur in parallel as opposed to sequentially, and the extent to which sub-tasks might be seen as a hierarchy.

2.2.1 Modularity of structure-building

Some theories assume language-specific skills and principles for syntactic structure-building. An early example is the Sausage Machine model proposed by Frazier and Fodor (1978), which is a two-stage model drawing on the parsing model proposed Kimball (1973) and in contrast to the Automated Transition Network model proposed by Kaplan (1972). In the first stage language input is analysed into phrasal chunks in a process which is independent of the previous sentence structure. These chunks are then passed to the second stage, which builds hierarchical clause structure. Application of the Minimal Attachment and Late Closure strategies in the first stage leads to phrasal chunks being as long as possible: potential reanalysis may then take place in the second stage. Thus the structure-building process can be captured in a module with two well-defined stages.

Frazier and Fodor's model was extremely influential, particularly the notion of a process involving initial construction followed by reanalysis. However, the empirically observed differences between languages and influence of lexical choice on the preferred attachment site of modifiers suggested that Frazier's Late Closure strategy was not universally applicable. In part to address these concerns, Frazier and Clifton (1996) put forward the Construal Hypothesis. Under this model, some elements of structure are attached immediately, whereas others are associated with a structural domain and then interpreted using structural and non-structural information. This complicates the situation for a modular view of processing: the first stage must be able to distinguish between arguments and modifiers, and the second stage must work not only on the basis of structural principles, but also have access to semantic contextual information for modifier attachment. Thus the modularity of structure-building is somewhat weakened.

Other theories assume that syntactic structure-building requires no distinct parser, but can be seen as part of general cognition. MacDonald et al. (1994) propose a model where the cognitive choices made during the building of syntactic structure are similar to those made during the selection of one lexical entry from memory to match a perceived word. Rather than specific principles governing attachment, lexical and contextual information on the combinatorial possibilities and their frequency is used to determine which of many possible options is chosen. Linguistic knowledge is lexically-based, and competence in structure-building is not contained within a modular process.

Probabilities, whether lexically specified and/or calculated from words in context, are also used in theories that do assume specific parsing processes or skills. Ford et al. (1982) proposed a process whereby phrases were built by a parser working through a valency template, with lexical knowledge about preferred alternations being used to govern the attachment of modifiers in ambiguous cases. More recently, theories have been proposed that assume the operation of a specific algorithm and then use probabilistic calculations to model time-course variations in attachment. Hale (2001) put forward the measure of Surprisal, and (Levy, 2008) proposed the measure of Entropy, both of which are Bayesian measures of probability of a word occurring given the previous sentential context. Levy demonstrates that surprisal and entropy are mathematically equivalent, and VanWagenen et al. (2014) show that the values calculated are dependent on modellers' assumptions of the parsing algorithm chosen and of the probability distribution of syntactic alternatives at choice points during the parsing process. With regard to modularity, probabilistic theories assume a language-specific parsing process which has access to the prior sentence structure, but do not need to specify detail of the structure-building process other than the assumed algorithm.

Other theories assume that a parsing algorithm applies, but use general cognitive principles to account for the variations in processing time-courses. This requires the detail of structure-building to be described. Lewis and Vasishth (2005) model a process in which lexical attachment takes place by retrieving the attachment site from a representation of syntactic structure in memory. General principles of memory activation including the strength of memory retrieval cues and interference from competitor memories are used to model the time-course of processing complex sentences. This approach is discussed further in Chapter 4.

2.2.2 Sequential organisation: one pathway or many?

Neurophysiological evidence suggests that many areas of the brain are involved during language processing, and that there is simultaneous activation of some areas. A review of neuroanatomical information is given by Price (2012), who synthesises twenty years of research reports on two techniques that identify brain activation: PET (positron emission tomography) and fMRI (functional magnetic resonance imaging). Price concludes that there is no single area of the brain that could be said to host "a language function" as such. Rather, particular production and comprehension tasks may be associated with activity in a particular area of the brain, but for processing as a whole these areas are distributed across the brain. This does not require processing tasks to take place simultaneously, and the distributed nature of processing is reflected in a variety of accounts of both written and spoken language processing. I will briefly consider in turn the models proposed by Hickok (2009), Hagoort and Indefrey (2014), Friederici (2012), and Bornkessel-Schlesewsky and Schlesewsky (2008, 2013) — from now on B-S&S.

The main focus of Hickok (2009) is on speech comprehension and production at the word level. He proposes a model of speech processing that has dorsal and ventral pathways in each hemisphere of the brain, working in parallel. These pathways are active in both comprehension and production, although for production more work is done by the dorsal pathway, linking to motor functions, whereas for perception more work is done by ventral pathways.

Hagoort and Indefrey's account (2014) of speech comprehension focuses on what they describe as 'unification', the bringing together of syntactic and semantic comprehension with context and other sources. They identify separate brain areas that are activated by complex syntax and semantics respectively, but do not go into detail about particular mechanisms by which the two streams are brought together.

Friederici (2012) also models speech comprehension. She reviews a number of studies and provides a fuller description of the brain areas involved in the processing of spoken language. Her model proposes dorsal and ventral channels for communication between these various brain areas, based on evidence about the presence of neural fibre systems. These neural fibre systems subserve a modular processing architecture, with the dorsal channel having separate steps for word recognition and morphosyntactic analysis; lexical phrase structure building; complex syntactic analysis; construction of representations of event or state predicates; matching of grammatical arguments to event or state participant roles; and integration with context (including prior discourse) and real world knowledge. The model assumes that the two channels are active simultaneously, although specific sub-tasks in the ventral route are not identified. A schematic diagram is presented in Figure 2.1.

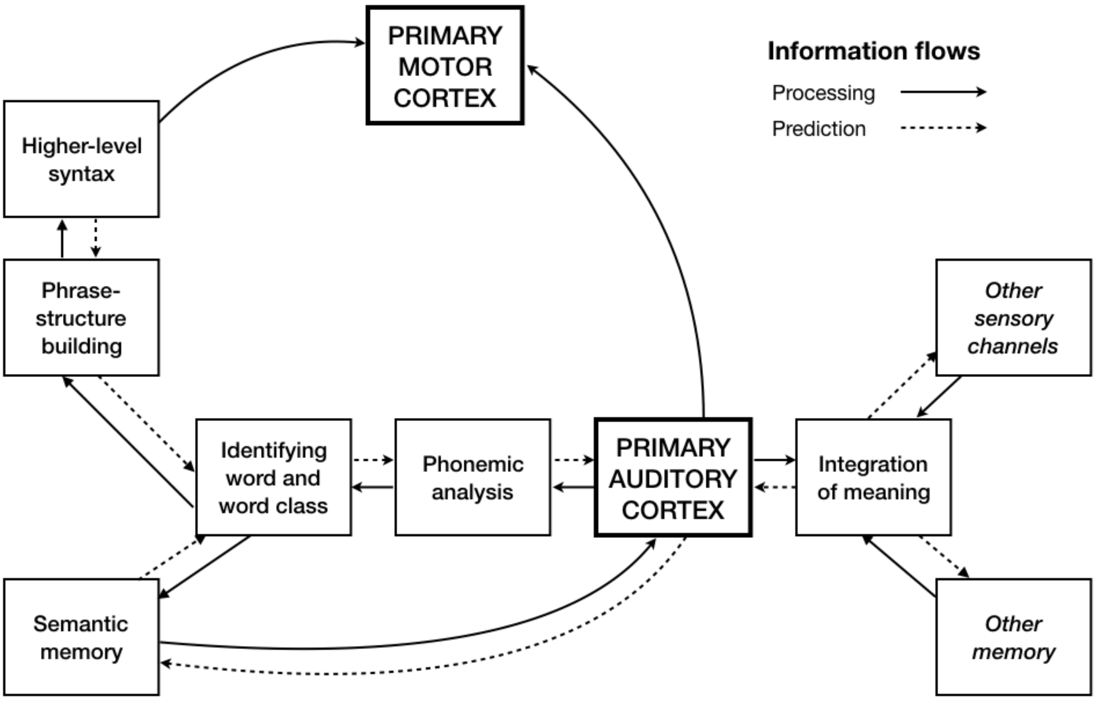


Figure 2.1: Schematic adaptation of Friederici's model (2012)

B-S&S (2008, also Bornkessel and Schlesewsky, 2006) proposed the Extended Argument Dependency Model (eADM), which considers in detail the process of syntactic structure building and, following Brouwer et al. (2012), grounding this in evidence from studies of N400 and P600 ERPs looking at syntax/semantics mismatches caused either by theta-role assignment or by stereotypically plausible/implausible arguments.

The eADM explicitly considers syntax and semantics, using VP/NP templates, lexical and morphosyntactic information, together with a combinatory process to match grammatical functions with semantic roles. However, other word categories beyond verbs and nouns are not considered. The model is divided into four phases, containing specific computational or retrieval tasks which the authors associate with specific brain areas and which align broadly with the processing stages identified by Friederici (2012).

A key element of the eADM is a cognitive input described as *lexical/associative se-*

mantic processing, which is distinct from word-by-word syntactic processing, and has a close relationship with the discourse environment. This stream interacts with syntactic processing in two modules, which the authors term *"plausibility" processing* and *generalised mapping*. These in turn are seen as one potential trigger of N400 and P600 ERPs respectively, Figure 2.2 shows a slightly simplified diagram of the model.

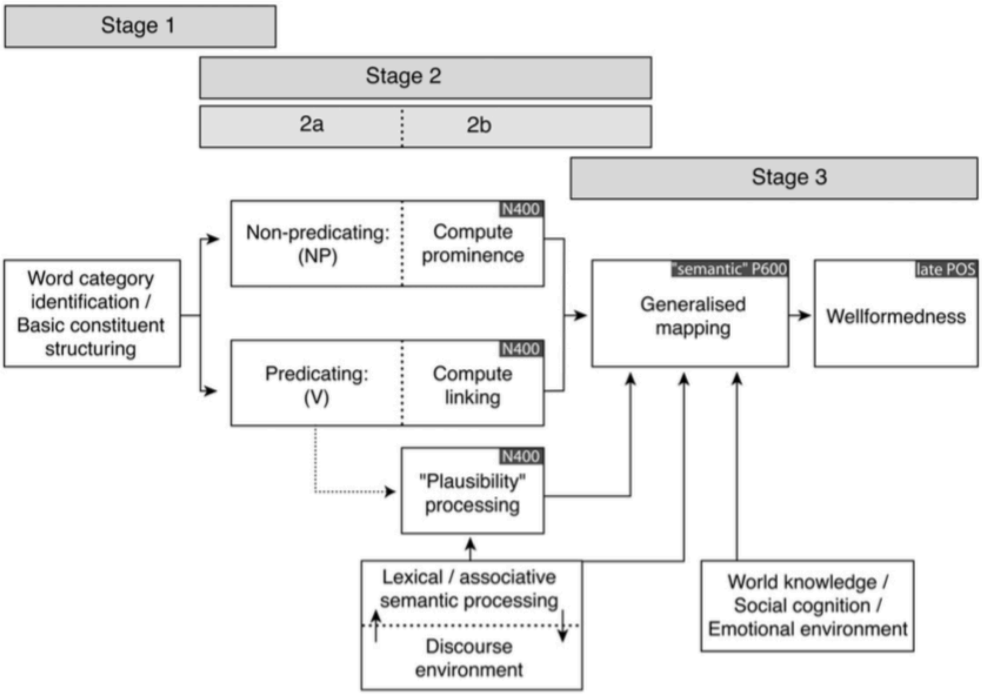


Figure 2.2: The Extended Argument Dependency Model, BS&S (2008, p.65)

In BS&S's subsequent work the model is revised as the Argument Dependency Model (2013), containing less detail about syntactic structure building, but including functionally-separated dorsal and ventral processing routes. The dorsal stream is assumed to be responsible for 'time-dependent' language processing, that is, processing where the linear order of information received has a direct impact on its comprehension. This includes prosody and constituent structure. Conversely, the ventral stream is responsible for 'time-independent' processing, the extraction of meaning. The authors represent the extraction of meaning by the unification of 'actor-event (AE) schemata'. This approach foregrounds an event representation and aims to take account of the cross-linguistically frequently-observed prominence of the argument bearing the agent role in a sentence. The AE-schemata provide the start of a formal approach to defining the discourse environment which, in the earlier eADM, informed the lexical/associative semantic processing.

A schematic diagram of the Argument Dependency Model is shown in Figure 2.3. The horizontal dashed line separates the dorsal from the ventral processing route; the vertical dashed line separates functions carried out in the temporal and parietal cortices from those carried out in the frontal cortex.

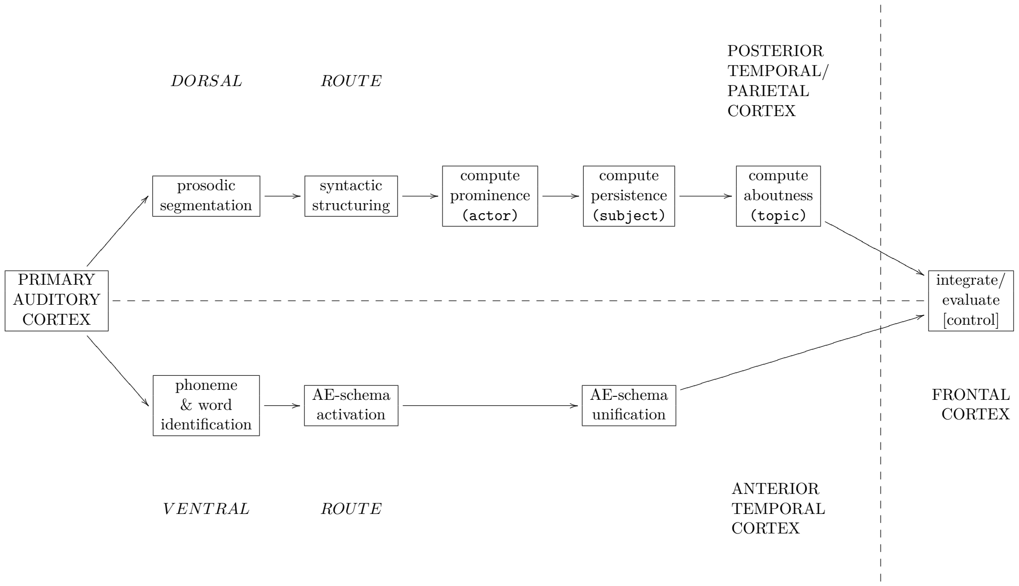


Figure 2.3: Diagram of the Argument Dependency Model, adapted from BS&S (2013)

The Argument Dependency Model, while less detailed, has some parallels to Friederici's proposal, although the work of word identification is split between the two channels rather than occurring before the point of separation.

2.2.2.1 Interactions between multiple processing streams

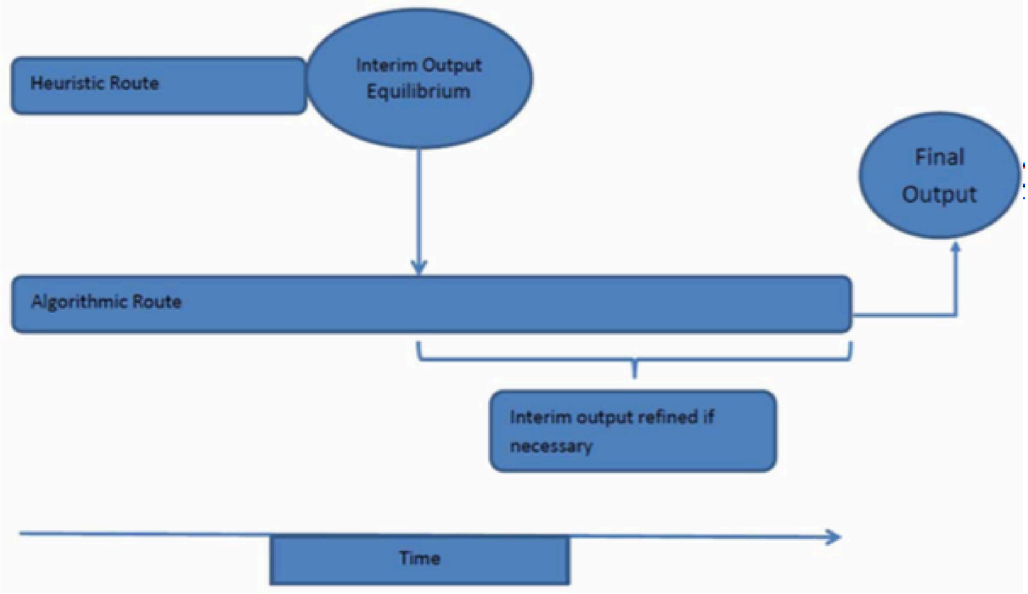
Although the theoretical models above take very different approaches, all four share a common element in the simultaneous operation of (at least) two structural pathways, reflected in a functional separation. Other authors have developed accounts that assume two functional streams — often termed 'syntax' and 'semantics' — and use the notions of competition or integration of these streams to explain processing effects observed through eye-tracking or ERP studies.

'Good enough' processing e.g. Ferreira et al. (2002); Ferreira and Patson (2007); Karimi and Ferreira (2015). "Good enough" processing theories assume that meaning can be derived from most linguistic input by means of heuristics, rather than requiring detailed syntactic analysis.

Karimi and Ferreira (2015) expand on the notion of Good Enough processing discussed earlier, and propose the Cognitive Equilibrium Hypothesis. Under this hypothesis, the brain has a state of equilibrium which is disturbed when a new stimulus is perceived which is not part of the existing context representation within the brain. The aim of cognitive processes is to restore equilibrium as quickly as possible by integrating the information derived from the new stimulus into the representation of context.

In Karimi and Ferreira's model (Figure 2.4), recognition of a word simultaneously triggers syntactic ('algorithmic') and semantic ('heuristic') comprehension processes, which operate in parallel. The semantic comprehension process provides information on meaning described as "interim output" which is then available for comparison with the output being generated by the syntactic process.

Figure 2.4: Language processing under the Cognitive Equilibrium Hypothesis, reproduced from Karimi and Ferreira (2015, p.7)



The authors claim that this model could account for the differences between fast and slow (online and offline) acceptability judgements, and that it accounts for phenomena such as differential processing times for underspecified anaphora resolution.

Monitoring Theory This theory, proposed by van Herten et al. (2006) and elaborated by others, looks at the interaction of a bottom-up syntactic process with a heuristically-driven semantic process linked to other perceptions and real-world knowledge.

Continued Combinatory Analysis Proposed by Kuperberg (2007), this is another model with separate syntactic and semantic streams. Under the theory, N400 ERPs relate to implausibility identified by either the syntactic or semantic stream, whereas P600 ERPs occur if there is conflict (i.e. one stream shows implausibility but the other does not).

Processing Competition The Processing Competition model (Hagoort et al., 2009; Kos et al., 2010) assumes that separate syntactic and semantic streams compete and that differing ERP effects arise from effort in the weaker of the two streams (at the time of conflict) to resolve the conflict.

2.2.2.2 Single stream models

Brouwer et al. (2012) compare the eADM, Monitoring Theory, Continued Combinatory Analysis, and Processing Competition with their proposed model, the Retrieval-

Integration model, which posits a single processing stream. They argue that the dual stream models described above fail to predict the observed pattern of N400 and P600 activation associated either with semantically implausible event participants, or with plausible participants in implausible thematic roles. Under the Retrieval-Integration model, they hypothesise that the aim of processing is to build a *mental representation of what is to be communicated* or MRC. This is updated as new lexical information is processed: the N400 response to a word reflects the degree of difficulty of memory retrieval relating to its lexical content, and the P600 reflects the processing load of integrating the computed representations arising from processing that word into the overall MRC. An increased processing load may arise because of plausibility factors, but may also reflect that a phrase or a clause has come to an end.

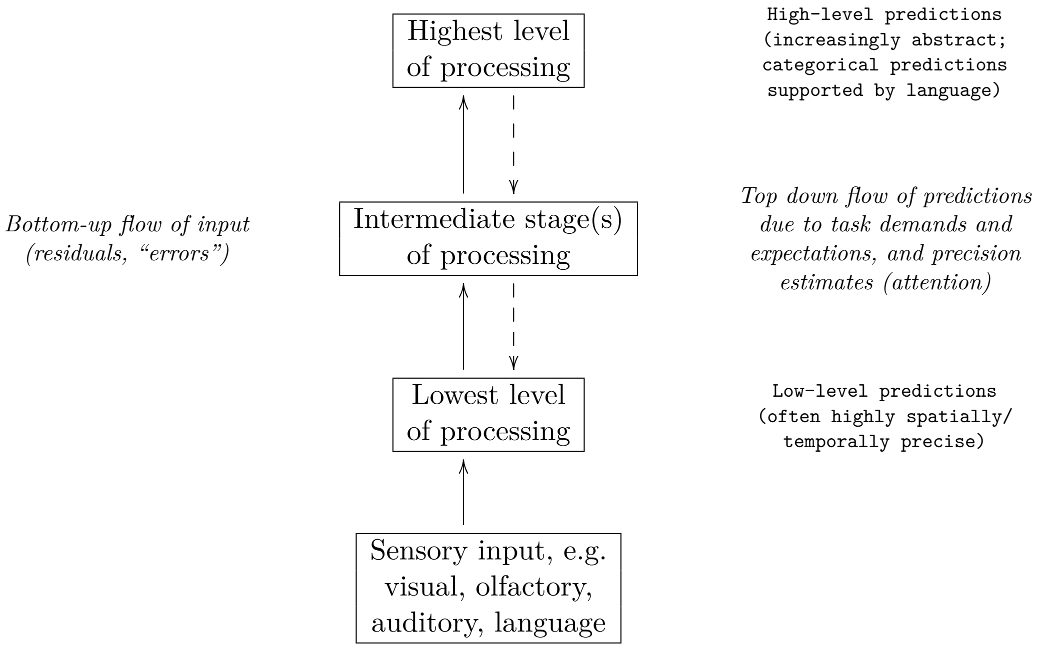
2.2.3 Hierarchical organisation

In Section 2.1, I presented evidence that context was able to influence the outcome of language processing, whether that context was derived from previously processed language input in the sentence, from the discourse, or from real world knowledge. There is also evidence that hearers may anticipate upcoming content, with several theories (e.g. Hickok, 2009; Indefrey, 2011; Friederici, 2012; Pickering and Garrod, 2013) assuming that prediction or forward modelling (a variety of terms are used) has at least a partial role in comprehension and possibly also in production, either as explicit prediction or as utterance planning. This may occur as a side-effect of memory retrieval processes, where retrieval of a particular piece of memory leads to the activation of associated knowledge, but may also result from hierarchically-organised processing.

Lupyan and Clark (2015) argue for a high level of integration between language processing and other cognition, but also for an ongoing contribution from both bottom-up inputs and top-down predictive modelling. Their proposal is that there are multiple stages of extracting meaning from any sensory input, including language, and that these are hierarchically organised from a highly abstract representation at the top, to a highly precise representation (e.g. temporal or spatial expectation of input) at the bottom. The outcome of processing at any given level is matched against an expected outcome derived from top-down predictions. Where this does not match, the resulting mismatch (described as 'unpredicted input', 'residuals' or '"errors"') is passed to a higher level for integration with more abstract predictions. This process is iterative, with each level refining its downwards prediction on the basis of top-down prediction and bottom-up 'unpredicted input', and also passing any residual 'unpredicted input' up the line to the next level. A schematic representation of this is given in Figure 2.5, adapted from Lupyan and Clark (2015).

Similarly, Kuperberg and Jaeger (2016) claim that a variety of phenomena regarding 'prediction' can best be understood if the language faculty is seen as a hierarchically organised series of processing tasks with the overall aim of comprehension being to infer

Figure 2.5: The predictive-processing account (adapted from Lupyan and Clark, 2015, p.280)



the speaker's intended meaning. They argue that this structure allows for multiple representations of meaning to be sustained, with Bayesian probabilistic adjustment of the representations as new information is received. Specifically their model builds on the tripartite parallel structure of language processing proposed by Jackendoff (2011).

What is interesting about Lupyan and Clark's and Kuperberg and Jaeger's approaches is that they seem congruent with models with a high degree of ongoing prediction (e.g. Levelt, 1989; Friederici, 2012; Hickok and Poeppel, 2004; Pickering and Garrod, 2013), and also with models where specific functions are linked to brain areas in a time-sequence (e.g. Indefrey, 2011; Hagoort, 2005) where each of the specific functions could be seen as an intermediate stage as in Figure 2.5.

There is some debate as to the extent to which grammar interacts with predictions based on cloze probability. Under the Semantic Attraction hypothesis proposed by Kim and Osterhout (2005), if an argument to a verb has a strongly associated role in the event depicted by the role, this association can override morphosyntactic information. For example, the thematic role of a *pizza* in relation to a *devouring* event is much more likely to be the theme rather than the agent. However, when experimental participants were presented with the sentence *The pizza was devouring the kids*, a high P600 ERP was recorded after *devouring*, rather than a high N400, which might have been predicted by the low plausibility and hence low cloze probability of a pizza *devouring* something. However, these effects can be modified very quickly, with Kos et al. (2010) demonstrating rapid reassessment of the level of plausibility of a theta-role violation in a prepositional adjunct to a verb.

In Friederici's model, discussed earlier, three forms of prediction are identified and related to specific areas of the brain. The first relates to syntactic category, where there is very rapid recognition of a word from an unexpected class (e.g. V instead of N). Friederici accounts for this by assuming that structure templates for lexical phrases are learned and stored locally. The other types of prediction are those associated with syntactic and with semantic priming. She proposes that syntactic information, such as the priming of active/passive voice or theta role assignment, travels along the dorsal neural channel, whereas semantic predictions such as a verb's likely arguments travel along the ventral pathway. Separately, Pickering and Garrod (2013) propose a role for prediction in speech production, in a model which closely couples mechanisms for production and comprehension.

2.2.4 Summary of processing accounts

Although the overall time-course of language comprehension is known, the exact nature of the sub-tasks within comprehension is still not clear. There is agreement between the theories discussed that meaning is not only built compositionally by computation from syntactic structure, but that there is also an element of semantically-driven processing. However, accounts vary as to whether this is a parallel, specifically linguistic semantic processing stream alongside syntactic processing, or whether general inferential processing takes place based on real world knowledge and stereotypical event representations, rather than the semantic content of an utterance. There are also differing accounts of the interface between different processing streams: while the P600 effect is recognised in many accounts as being linked to semantic integration, exactly what that entails varies between accounts. There is also evidence that linguistic context and real world knowledge provide some kind of feedback during the processing of an utterance, so that the early stages of an utterance shape the probabilities for interpretation of the later content.

In Chapters 3, 5 and 6 I draw on the theoretical assumptions of Frazier and Fodor's strategy-based models and MacDonald et al.'s probability-based model to guide my assumptions about the first stages of building syntactic structure. The sequential models of the overall processing pathway inform the proposal in Chapter 5 about how LFG levels of representation might map onto time-sequenced processing tasks.

2.3 Prosody and language structure

The review of research on prosody and language structure by Wagner and Watson (2010) gives, in the authors' own terms, a rough definition of prosody as "a level of linguistic representation at which the acoustic-phonetic properties of an utterance vary independently of its lexical items" (p. 905). The acoustic correlates of prosody are duration, pitch, and intensity. These combine to divide utterances into phrases of different lengths, the inventory of which may vary between languages (Jun, 2005b) but which are proposed to be organised hierarchically according to the Strict Layer Hypothesis (Selkirk, 1984) such that the presence of a prosodic boundary in a higher layer entails the presence of a boundary in a lower layer. Although the realisation of boundaries varies between languages, Langus et al. (2012) found that learners of a second language were able to use unfamiliar prosodic cues from their second language to identify syntactic structures, even where these were different from the boundary cues used in their native language.

The Implicit Prosody Hypothesis (Fodor, 2002) states that "in silent reading, a default prosodic contour is projected onto the stimulus, and it may influence syntactic ambiguity resolution. Other things being equal, the parser favors the syntactic analysis associated with the most natural (default) prosodic contour for the construction." Fodor claims that because the placement of prosodic phrase boundaries is affected universally by constituent length, but also by language-specific rules which may have be linked to syntactic structure, it can account for cross-linguistic differences in attachment preference. It is also open to experimental manipulation by varying phrase length or by introducing punctuation.

Prosodic phrase boundaries are not equal: for example, Jun (2005a) identifies that Korean has Intonational Phrase and Accentual Phrase boundaries. Fine-grained explorations of the relative strength of prosodic phrase boundaries have shown that this can influence hearer's interpretations of an utterance (Carlson et al., 2001; Clifton et al., 2002). However, this level of granularity is frequently not considered in research, and for the purposes of this project I assume that the differences between types of prosodic phrase boundary are not material. From here onwards, following Fodor (2013), the symbol || indicates a prosodic break above the level of a prosodic word.

2.3.1 Prosody and attachment preference

Prosodic information is used very early as part of building initial structure, if it is available (Marslen-Wilson et al., 1992; Snedeker and Trueswell, 2003; Watson and Gibson, 2005). It is used by hearers to create elements of hierarchical structure. For example, Kjelgaard and Speer (1999) looked at sentences with an initial modifying clause, where the position of a prosodic boundary could be varied (23).

(23) a. When Roger leaves || the house is dark.
b. When Roger leaves the house || it's dark.

They found that hearers used prosody to assign the end of the modifying clause, with the constituent following the boundary preferentially being taken as the subject of the matrix clause.

Schafer et al. (2000) also explored the impact of prosody on the early or late closure of an initial sentence-modifying clause, using an experimental paradigm that collected naturally-generated speech samples and then presented these to hearers. They found that speakers and hearers used prosody to disambiguate with the patterns seen in (24) whether or not the context surrounding the sentence removed potential ambiguity.

The presence of a prosodic boundary can also guide hearers' preferences in attachment of a dependent to a head, where more than one attachment site is possible. Snedeker and Trueswell (2003) compared the two sentences in (24).

(24) a. *Tap the frog || with the flower.*
b. *Tap || the frog with the flower.*

For both speakers and hearers, version (24a) was predominantly associated with a high attachment, which can be paraphrased "use the flower to tap the frog," whereas version (24) was predominantly associated with the low attachment, which can be paraphrased "tap the frog that has the flower."

The above examples used prosodic boundaries to establish the presence of a syntactic boundary in situations where there was some ambiguity. Watson and Gibson (2005) found that in the absence of any ambiguity, processing was disrupted where a prosodic boundary interrupted a head-dependent relationship. They argue that a prosodic boundary is interpreted by listeners as marking the end of a constituent, which has the effect of a separate attachment point being sought for the material that is processed after the break.

2.3.2 Prosody and complex syntax

Prosodic boundaries interact with complex syntax, and this can facilitate or inhibit ease of processing, depending on boundary placement. Hwang and Steinhauer (2011) investigated the effect of prosody on the processing of the complex Korean sentences in (25).

(25) a. Short subject, NP-DAT ambiguous between matrix and embedded clause.

phikules-i lopin-eykey phwuwu-ka ttacwu-n pelcip-ul unkunsulccek
Piglet-NOM Robin-DAT Pooh-NOM pick-REL honeycomb-ACC stealthily
phalapelyessta
sold

"Piglet stealthily sold [the honeycomb [that Pooh picked for Robin]]." OR

"Piglet stealthily sold Robin [the honeycomb [that Pooh picked]]."

b. Short subject disambiguated by second NP-DAT, potential garden path.

phikules-i lopin-eykey phwuwu-ka ttacwu-n pelcip-ul thike-eykey
Piglet-NOM Robin-DAT Pooh-NOM pick-REL honeycomb-ACC Tigger-DAT
phalapelyessta
sold

"Piglet sold Tigger [the honeycomb [that Pooh picked for Robin]]."

c. Long subject, NP-DAT ambiguous between matrix and embedded clause.

akitwayci phikules-i lopin-eykey phwuwu-ka ttacwu-n pelcip-ul
little Piglet-NOM Robin-DAT Pooh-NOM pick-REL honeycomb-ACC
unkunsulccek phalapelyessta
stealthily sold

"Little Piglet stealthily sold [the honeycomb [that Pooh picked for Robin]]."
OR

"Little Piglet stealthily sold Robin [the honeycomb [that Pooh picked]]."

d. Long subject disambiguated by second NP-DAT, potential garden path.

akitwayci phikules-i lopin-eykey phwuwu-ka ttacwu-n pelcip-ul
little Piglet-NOM Robin-DAT Pooh-NOM pick-REL honeycomb-ACC
thike-eykey phalapelyessta
Tigger-DAT sold

"Little Piglet sold Tigger [the honeycomb [that Pooh picked for Robin]]."

Sentences (25a) and (25c) are ambiguous. If the second noun *lopin-eykey* 'Robin-DAT' is taken to be the left edge of an embedded clause, they have the reading *(Little) Piglet stealthily sold the honeycomb that Pooh picked for Robin.* However, if the second noun is part of the matrix clause and the left edge of the embedded clause is only recognised when a second nominative-marked noun *phwuwu-ka* 'Pooh-NOM' is encountered, the reading is *(Little) Piglet sold Robin the honeycomb that Pooh picked.*

By contrast, sentences (25b) and (25d) are unambiguous. Because there is a dative-marked noun in both the matrix and the embedded clauses, the reading is *Little Piglet sold Tigger the honeycomb that Pooh picked for Robin.* However, the left edge of the embedded clause is still not clear: if it is only recognised when the second nominative-marked constituent is encountered, a reanalysis is triggered because the word *thike-eykey* 'Tigger-DAT' suggests that there are two dative-marked constituents in the matrix clause. There is thus the potential for these variants to be garden path sentences.

A previous speech elicitation study (Hwang and Schafer, 2009) suggested that where the initial NP was lengthened by adding a modifier, as in sentences (25c) and (25d), there was more likely to be a prosodic break before the first dative-marked noun *lopin-eykey* 'Robin-DAT'. As a result, the authors predicted that it would be easier to analyse this NP as part of the embedded clause in (25d) than in (25b), and hence any P600 effect relating to a garden path reanalysis after *thike-eykey* 'Tigger-DAT' should be lower. The experimental results were in line with this prediction, and so were taken in support of the implicit prosody hypothesis for written language, and the role of prosodic breaks in facilitating the recognition of clause boundaries.

Interactions between prosody and complex syntax have also been reported for English. For example, Fodor (2013) claims that although multiply centre-embedded sentences such as (26) cause severe parsing difficulty, a particular pattern of prosodic breaks (27) allows them to be parsed successfully (Fodor's examples (23) and (24), p.222).

(26) The elderly Frenchwoman || that the man the girl loved || met on a Mediterranean cruise || died last year in Maine.

(27) The elderly and eccentric French doctor || that the man the girl loved met || died last year in northern Maine.

2.3.3 Prosody and the contravention of syntactic constraints

Kiaer (2011, p.172) notes that prosody can license “unusual constituent” phenomena, such as the fronting of two arguments of an embedded ditransitive verb, which is often claimed to be ungrammatical in Korean. Usually, only one argument of an embedded ditransitive verb can be extracted to become a sentence-initial topic, as shown in (28) or (29). To make the embedding and extraction clear, the glosses carry prefixes indicating the grammatical function of each constituent, together with the clause in which it sits.

(28) OBL$_2$*:suphai-hanthey* SUBJ$_1$*:Kim-kemsa-nun* SUBJ$_2$*:Park-kica-ka*
OBL$_2$:spy-DAT SUBJ$_1$:Kim-prosecutor-TOP SUBJ$_2$:Park-reporter-NOM
ADV$_2$*:mollay* OBJ$_2$*:passwordu-lul* PRED$_2$*:malhayssta-ko* PRED$_1$*:mitesseyo*
ADV$_2$:secretly OBJ$_2$:password-ACC PRED$_2$:said-COMP PRED$_1$:believed
“It was to the spy that Prosecutor Kim believed that reporter Park secretly told the password.”

(29) OBJ$_2$*:passwordu-lul* SUBJ$_1$*:Kim-kemsa-nun* SUBJ$_2$*:Park-kica-ka*
OBJ$_2$:password-ACC SUBJ$_1$:Kim-prosecutor-TOP SUBJ$_2$:Park-reporter-NOM
OBL$_2$*:suphai-hanthey* ADV$_2$*:mollay* PRED$_2$*:malhayssta-ko* PRED$_1$*:mitesseyo*
OBL$_2$:spy-DAT ADV$_2$:secretly PRED$_2$:said-COMP PRED$_1$:believed
“It was the password that Prosecutor Kim believed that reporter Park secretly told the spy.”

However, if the first two words are contained in a single Accentual Phrase, the sentence becomes acceptable, as is shown in (30). Note that in all three of Kiaer’s examples, the morphological marking does not match the information structure: the fronted topic arguments carry their case marking, and the matrix subject is marked as the sentence topic.

(30) OBJ$_2$*:passwordu-lul* OBL$_2$*:suphai-hanthey* SUBJ$_1$*:Kim-kemsa-nun*
OBJ$_2$:password-ACC OBL$_2$:spy-DAT SUBJ$_1$:Kim-prosecutor-TOP
SUBJ$_2$*:Park-kica-ka* ADV$_2$*:mollay* PRED$_2$*:malhayssta-ko* PRED$_1$*:mitesseyo*
SUBJ$_2$:Park-reporter-NOM ADV$_2$:secretly PRED$_2$:said-COMP PRED$_1$:believed
“It was the password to the spy that Prosecutor Kim believed that reporter Park secretly told.”

Thus Kiaer shows how prosody can reduce the unacceptability of a syntactic constraint being contravened.

2.3.4 Summary of prosodic factors

In summary, although there are many complexities around the description and detailed classification of prosodic boundaries, there is a clear tendency for a prosodic boundary to be interpreted as the edge of a constituent. Thus if a prosodic boundary reinforces the identification of a syntactic boundary at an otherwise ambiguous point, it may serve to facilitate the processing of complex syntax. Conversely, a misalignment between prosody and syntax may reduce the ease of processing by disrupting local syntactic relationships.

2.4 Representing language structure

The majority of psycholinguistic work represents the hierarchical structure of language as some variant of a constituent structure tree, and describes syntactic constraints in terms of proposals by Chomsky, such as his Binding Theory (Chomsky, 1981), referred to by Sturt (2003), Kazanina et al. (2007) and others. However, there are challenges inherent in using Chomsky's theories to develop a processing account. One long-recognised challenge (Fodor, 1978) is that many syntactic constraints within the theories operate during the process of generating a surface form from a deeper representation, rather than at the surface form itself. Ferreira (2005) discusses the difficulties of relating the Minimalist Program (Chomsky, 1995) to psycholinguistics, noting that "the model derives a tree starting from all the lexical items and working up to the top-most node, which obviously is difficult to reconcile with left-to-right incremental parsing." (Ferreira, 2005, p. 370). She also notes the challenge that notions such as "spell-out" cause for the reanalyses necessary to reanalyse garden path sentences with an object-subject ambiguity.

2.4.1 Incremental approaches to Chomskyan syntax

Attempts have been made to incorporate incremental elements into Chomskyan theory, for example by Phillips (2003) and Stabler (2010, 2013). These allow phrase-structure grammars that are compiled using Chomskyan syntactic constraints to be used with a parsing algorithm, thus allowing incremental growth of structure whilst maintaining fidelity to the theory. Examples of authors following such an approach include VanWagenen et al. (2014) and Hale (2016): the impact of structural factors on processing can thus be modelled. However, there is no relationship between syntax and semantics inherent in this approach, and so further assumptions must be made to incorporate semantic effects on processing.

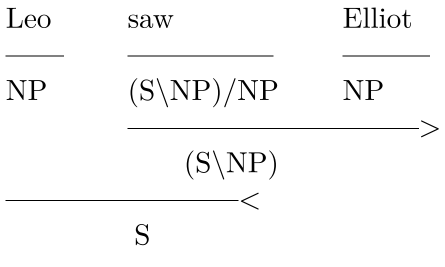
Some analyses, including but not limited to Chomskyan approaches, can also require the presence of empty terminal nodes in a constituent-structure tree. In these accounts, LDDs are seen as involving extraction of an element from a constituent structure and moving it elsewhere, with a trace of the extracted element remaining as a morphologically empty node. This is a highly intuitive analysis and the terminology *filler-gap dependency* is often used to describe the process of creating an LDD even within theories that do not permit empty categories.

Morphologically empty nodes are problematic for an incremental theory because their presence or absence at a given point is impossible to measure. Some authors, e.g. Pickering and Barry (1991), have suggested that it is not necessary to assume empty categories in order to account for processing data, and that a more parsimonious account would not include this assumption. Pickering and Barry argued that a Categorial Grammar framework (Ajdukiewicz, 1935; Bar-Hillel, 1953) might be better suited to psycholinguistic accounts. However, their claims regarding empty categories and their proposal to move away from the Chomskyan framework were met with counterarguments. Gibson and Hickok (1993) argued that it was possible to account for Pickering and Barry's conclusions using an algorithm that assumes the early projection of empty categories, and therefore Pickering and Barry's conclusion requiring a different grammatical approach was too strong. Thus for Gibson and Hickok, the question of whether it is less parsimonious to include empty categories in an account appears to be moot.

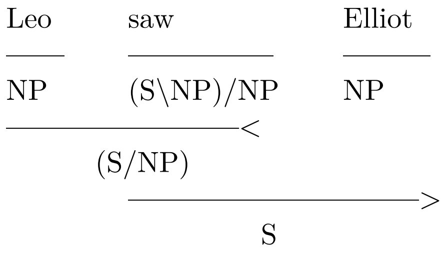
2.4.2 Categorial grammars

Categorial Grammars such as Combinatory Categorial Grammar, CCG, (Steedman, 2000) use a flexible notion of constituency where each word is associated with a type (e.g. NP for proper nouns, (S\NP)/NP for transitive verbs). Type specifications give information about combination: the type S/NP means "if a constituent of my type is followed by a constituent of type NP, the resulting constituent has type S" whereas type S\NP means "if a constituent of my type is preceded by a constituent of type NP, the resulting constituent has type S". Types are combined by functional application using either forward application > or backward application < to the constituents. In forward application the function is applied forwards from the constituent on the left, and so combines categories X/Y and Y (in left-right order), whereas in backward application the function is applied backwards from the constituent on the right, and so combines categories Y and X\Y (also in left-right order). In example (31) replicated from Phillips (2003, p. 79), there are two methods by which the sentence *Leo saw Elliot* can be parsed to generate the root sentential type S.

(31) a.

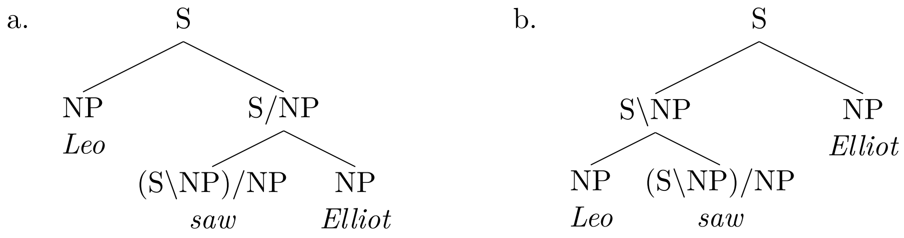


b.



Constituent types are also associated with propositional content using lambda terms, giving a transparent link between sentence structure and propositional content. Although the final representation is not a tree, a hierarchical structure can be built by working backwards through functional composition. Thus the analyses in (31) could also be represented as in (32).

(32)

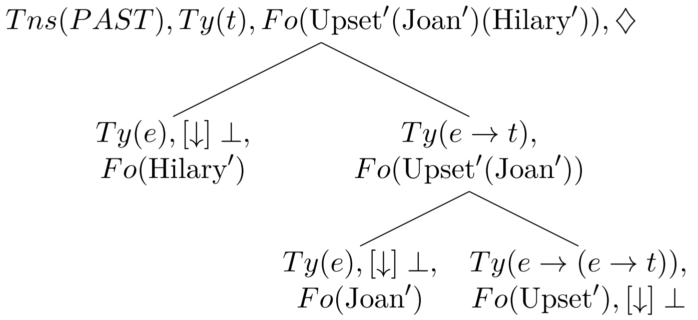


CCG allows structure to be built incrementally with repeated application of functions. However, as Phillips points out, the structures formed in this way are left-branching as in example (32b) and so the resulting constituent structure may not match the constituents required to effectively describe syntactic constraints on co-reference or LDDs.

2.4.3 Dynamic Syntax

Dynamic Syntax (Cann et al., 2005) also uses a tree to represent structure, but the tree contains semantic, rather than syntactic relationships. Example (33), reproduced from Cann et al. (2005, p. 55) shows the final analysis for the sentence *Hilary upset Joan.* Each node of the tree is decorated with semantic information including as a minimum a type value, e.g. $Ty(e \to t)$, and a partial proposition expressed in the Dynamic Syntax *Formula* language, e.g. $Fo(\text{Upset}'(\text{Joan}'))$. During parsing, these decorations are augmented to include type requirements of incomplete nodes or subtrees, e.g. the requirement $?Ty(t)$ on the root node indicates that the node will be complete when its proposition is evaluable to a term of type t. Nodes can also carry constraints such as $[\downarrow] \perp$ which indicates "my daughter node is false", in other words, this node must be a terminal node on the tree.

(33) Hilary upset Joan.



Syntax is described by the path of the pointer $\Diamond$ through the tree as nodes are added and decorated in accordance with lexical specifications and previously-added constraints. Lexical specifications also govern the pointer's path through the tree, indicating an expectation of where information will be added next. However, the theory allows for unexpected input to be parsed, with additional structure created as necessary to allow lexical input to be processed. The theory claims that the ordering constraints and lexical specifications in different languages result in different patterns of tree growth, but that the final representation of meaning is universal between languages.

Dynamic Syntax thus also provides a transparent link between form and meaning, but the requirements of the lexical specification mean that the process of initially adding a word into a representation can be quite complex. Example (34) shows the lexical specification for *upset* used in the analysis in (33), reproduced from Cann et al. (2005, p. 48).

(34) *upset*

```
IF     $Ty(e \to t)$                                                  Predicate trigger
THEN   go($\langle\uparrow_1\rangle ?Ty(t)$);                          Go to propositional node
       put($Tns(PAST)$));                                             Tense information
       go$\langle\downarrow_1\rangle ?Ty(e \to t)$;                    Go to predicate node
       make($\langle_1\rangle$);                                       Make functor node
       put($Fo(\text{Upset}'), Ty(e \to (e \to t)), [\downarrow] \perp$);   Annotation
       go($\langle\uparrow_1\rangle$);                                 Go to mother node
       make($\langle\downarrow_0\rangle$);                             Make argument node
       go($\langle\downarrow_0\rangle$);                               Go to argument node
       put($?Ty(e)$)                                                  Annotate
ELSE   Abort
```

The specification requires four distinct pointer movements, indicated by the instruction `go`, and five distinct amendments to the tree, indicated by the instructions `put` and `make`, all of which must be complete before any propositional evaluation can take place and the next word can be added to the representation. The procedure is rigorously incremental, but the ordering of tasks within the tree-building algorithm privileges the syntactic process above calculations of meaning at each word. It is unclear where interactions between semantics and syntax, such as the plausibility effects occurring after initial structure building described by Sturt (2003), might take place in this theory.

2.4.4 Summary of syntactic representations

Theories of grammar have been proposed that either are able to provide incremental analyses of language data, or which use the incremental growth of sentence structure as the organising principle of their account. This makes it easier to build theoretical bridges between grammar and processing. However, the theories' internal assumptions about the order in which syntactic and semantic information is brought to bear on an analysis do not necessarily align with empirical data about the staging of different processing tasks. This suggests that there is value in considering how other grammar theories might be extended to offer incremental accounts, and motivates the proposal of an incremental account of LFG in Chapter 3.

2.5 Summary

This chapter reviewed the evidence on the relationship between language structure and processing, the pathways that have been proposed for language comprehension, and some proposals regarding the representation of hierarchical structure. There is evidence that syntactic structure and syntactic constraints interact with processing, but that non-syntactic factors also influence processing at choice points: other than at the very earliest stages of processing, syntactic constraints on LDDs and co-reference are violable.

The impact of prosody on language processing can be considerable, with the positioning of prosodic boundaries relative to constituent edges capable of facilitating or disrupting processing. Much of the experimental evidence reported in Section 2.1 made use of written stimuli and did not take prosody into account, although the implicit prosody hypothesis suggests that prosody is always present for comprehenders even when reading stimuli. Including prosody in the representation of language structure is therefore one way to enhance our accounts of processing. The model presented in Chapter 6 is capable of processing prosodic breaks as part of lexical input, and suggests how the two may interact.

Representations of language structure often focus on constituent-structure trees, and the relationship between the input string, the operation of syntactic constraints, and representations of language meaning is not always transparent. Where explicitly incremental approaches to grammar are proposed, such as CCG or Dynamic Syntax, these may not satisfactorily allow the interactions between semantics and syntactic constraints that have been demonstrated experimentally.

Thus, in seeking to model the role of grammar in language processing, it is the very earliest stages of structure-building where the relationship between grammar and processing is least complicated by the need to build other factors into the model. Accordingly, a productive route for a researcher seeking to model language processing is to choose an incremental theory of grammar that offers a consistent approach along the many steps between language form and language meaning and then, as a starting point, to use that grammar to focus on the first stages of building a syntactic representation. In Chapter 3 I set out a treatment of incrementality in LFG, which articulates syntactic constraints on structure growth in terms of established theoretical principles. The discussion focuses on the earliest stages of structure-building but is extensible by future work to cover later stages of the language processing pathway.

Chapter 3

An incremental theory of LFG

In the first part of this chapter I introduce Lexical Functional Grammar, giving a description of LFG's formal architecture, including the pivotal role of functional structure in specifying the relationship between form and meaning. I focus on the areas of the theory within the scope of this thesis, indicating where other elements of the grammar can be considered in future work.

In the second part of the chapter, I explain how LFG can be articulated as an incremental theory, which can then form the basis of an exploration of the role of grammar in language processing. This begins with a discussion of the ability of LFG to provide detailed analyses of language fragments, followed by the theoretical extension of this to provide incremental analyses of a sequence of increasingly longer fragments building word-by-word from the start of an utterance. Having done this, I propose some aims for an incremental grammar theory, using the characteristics of incremental growth set out in Asudeh (2012) and work through LFG modules and interfaces to show how universal and language-specific constraints guide the process of structure-building. I also consider the impact of an incremental approach on structure-sharing, which underpins the LFG account of long-distance dependencies.

3.1 The formal structure of an LFG analysis

In this section I describe the formal structure of Lexical Functional Grammar, including the underlying theoretical assumptions that I will be using in my model. Within the LFG literature, some elements of the grammar are commonly assumed, whereas other elements have competing approaches. I will make reference to competitor approaches that I have not assumed, but will only discuss my assumptions in detail. I will also mention — but not discuss in detail — areas of the theory that could be added to my model but that are outside the scope of this project.

3.1.1 Modular architecture

LFG seeks to account for systematic correspondences between form and meaning using a modular architecture that allows language-specific patterns and constraints to be described whilst at the same time identifying universal patterns and constraints that obtain cross-linguistically despite great differences in the surface form of languages. Bresnan et al. (2016) state three principles that inform the architecture of LFG: variability, universality, and monotonicity.

Variability: This principle states that "external structures vary across languages" (*ibid.* p. 41), with c-structure as the LFG representation of external syntactic structure, and elements of that structure being subject to ordering constraints in linear precedence, dominance, and category.

Universality: This principle states that "internal structures are largely invariant across languages" (*ibid.* p.43), with f-structure as the LFG representation of internal syntactic structure.

Monotonicity: This principle requires that the correspondence between an utterance's external structure and its meaning is monotonic, in that the greater the amount of surface structure that is available, the more detailed and specific the representation of meaning in the grammar. There are no transformations. Although some authors, (e.g. Bresnan et al., 2016), make use of empty categories that occupy the terminal nodes of a constituent-structure tree, other accounts assume that only expressed lexical items can project terminal tree nodes, thus removing the need to assume any covert elements. In LFG, this correspondence "is designed to preserve inclusion relations between the information expressed by the external structure and the content of the internal structure." (*ibid.* p.43).

The separation of external and internal linguistic structure is reflected in a modular architecture that describes the relationship between language and meaning as an ordered set of levels of representation, and a set of projection functions that govern the mapping between one level of representation and the next. The grammaticality of an utterance is determined by the conformity of its analysis with well-formedness constraints that apply at particular levels of representation. The set of constraints includes both universal and language-specific elements.

The effect of modularity is that only the linguistic information that is pertinent to a particular level of representation, appears in that level. Each level of representation constitutes the domain of a projection function whose range is the domain of the next level of representation, until the final level of the analysis is reached. Analyses are declarative and monotonic: there is no concept of movement during an analysis; a given element of any level of representation must contain all of the information that is supplied by any elements that are subsidiary to it; and the levels of representation are mutually constraining.

A number of different proposals have been made as to the levels of representation that are required in LFG and the mapping relationships between them, with authors seeking the most parsimonious architecture that can account for language data and maintain LFG's modularity. The architecture I assume is shown in Figure 3.1, with the arrows representing projections from one level of representation to the next.

$$\begin{tikzcd}
\text{lexicon} \arrow[r] & \text{s-string} \arrow[r, "\pi"] & \text{c-structure} \arrow[r, "\phi"] & \text{f-structure} \arrow[r, "\sigma"] & \text{s-structure}
\end{tikzcd}$$

Figure 3.1: The architecture assumed for my model

I follow Asudeh and Giorgolo (2012) in that there is no separate representation of argument structure. My focus is on the building of syntactic structure and so in the subsequent thesis I will primarily consider c- and f-structures. I include s-structure in the initial description because f-structure is the domain of the function that projects s-structure as its range. Although I will be considering the role of prosodic boundaries in syntactic disambiguation, I will take the simplifying assumption that a prosodic break is included as an element of the s-string, following Bögel et al. (2009). A more detailed incremental treatment of prosody, including the richer structural representation proposed by Dalrymple and Mycock (2011) and Mycock and Lowe (2013) is left for future work.

I now introduce each of the levels of representation in more detail, including their formal definitions, the constraints that apply within them and, where appropriate, those elements of a level that are language-specific. Following that, I discuss the projection functions[1] that describe the correspondences between levels.

3.1.2 The lexicon

Each language has a lexicon which includes all the words of the language. The lexicon is a set of lexical specifications that minimally comprise an s-form, c-structure information, f-structure information, and a meaning constructor. Two example lexical specifications are shown in (35), one from English (35a), and one from the Australian language Wambaya (35b) adapted from Nordlinger (1998, p. 64).

(35) a. *kissed* V $(\uparrow \textsc{pred}) =$ ‘kiss $\langle \textsc{subj}, \textsc{obj} \rangle$’
$(\uparrow \textsc{tense}) = \textsc{pst}$

$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg1})$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg2})$

$\lambda y \lambda x \lambda e.kiss(e) \wedge agent(e) = x \wedge patient(e) = y :$
$(\uparrow_\sigma \textsc{arg2}) \multimap (\uparrow_\sigma \textsc{arg1}) \multimap (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

---

[1] I use *projections functions* as an umbrella term following widespread practice in the LFG literature. As Lowe (2016) points out, if lexical sharing (Wescoat, 1987, 2002) is assumed, π must be a relation rather than a function.

b. *alaji* N (↑ PRED) = 'boy'
(↑ GEND) = M
(OBJ ↑)
(↑ CASE) = ACC

$\lambda x.boy(x) : (\uparrow_\sigma \text{VAR}) \multimap (\uparrow_\sigma \text{RESTR})$

In the specification at (35a), *kissed* is the s-form, the written form of the word[2]. Each lexical specification has only one s-form, but one s-form may be included in several lexical specifications. The c-structure information comprises the category label V, which in English is a lexical, projecting category. The first block of information in the right-hand column of the lexical specification is f-structure information. For *kissed*, this states that the word contributes values for the PRED and TENSE attributes of the relevant f-structure. The second block of information links f-structure and s-structure, stating that the f-structure must contain SUBJ and OBJ attributes, and that these correspond to the ARG1 and ARG2 attributes of s-structure respectively. The third block of information is the meaning constructor, which is used to derive propositional content using Glue Semantics (Dalrymple, 1999).

The specification at (35b) also contains f-structure information and a meaning constructor, although it does not contribute any information about the link between f-structure and s-structure. The f-structure information includes the inside-out functional constraint (OBJ ↑) which indicates that the f-structure for which *alaji* contributes the PRED value bears the grammatical function OBJ in relation to the PRED value of the f-structure that immediately contains it. F-structures are introduced in detail in Section 3.1.5: for the moment it is sufficient to note that lexical specifications may constrain the relationship between f-structures as well as contributing information to a particular f-structure. The meaning constructor again provides information to allow propositional content to be derived using Glue Semantics[3].

3.1.3 The s-string

The s-string holds the language input which is being analysed. It is an ordered set of s-forms drawn from the lexicon.

3.1.4 C-structure

C-structure, short for constituent structure, represents the hierarchical relationship between the words in the s-string, grouped into constituents according to the grammar of the language.

---

[2] The phonological representation of a word, its p-form (Dalrymple and Mycock, 2011), is also included in a full lexical specification but a detailed consideration of prosody is outside the scope of this thesis.

[3] For a more detailed introduction to Glue Semantics in meaning constructors see Dalrymple (2001, Ch. 8).

The formal definition of c-structure is given by Kaplan (1995, p. 10), repeated here as example (36).

(36)
- a. N: set of nodes
- b. L: set of category labels
- c. M: Mother function, N→N
- d. $\prec$: precedence relationship, $\prec \subseteq$ N×N
- e. λ: labelling function, N→L

In this definition, a c-structure consists of a set of nodes N and a set of category labels L. The mother function M maps a node onto the node that immediately dominates it in the tree. A node n is a root node of the tree if $\mathrm{M}(n) = \emptyset$. For clarity in writing formulae, I will where appropriate follow the convention that the mother of a node $*$ is represented as $\hat{*}$.

In Section 3.2 I will make use of the c-structure dominance relationship, which is the transitive closure of the mother function M. A node n_1 dominates another node n_2 if either $\mathrm{M}(n_2) = n_1$ or there is a node x such that $\mathrm{M}(x) = n_1$ and x dominates n_2.

There is a partial ordering relationship $\prec$ between nodes such that $n_i \prec n_j$ entails that n_i precedes n_j. The precedence relationship is transitive: $n_1 \prec n_2 \wedge n_2 \prec n_3 \Rightarrow n_1 \prec n_3$. Although not explicitly stated by Kaplan, the precedence relationship is such that a node vacuously precedes and is preceded by itself.

Terminal nodes in the tree branches are projected from the s-string by the mapping relation π. Each head node projected by the π relation has a category, determined by the lexical specification of the projecting word. Categories distinguish subsets of the lexicon that share particular syntactic characteristics (e.g. category V representing verbs, or category N representing nouns). They are drawn from a language-specific category set with cross-linguistic generalisations about categories[4]. However, because c-structure is language specific, there is no requirement or expectation that all categories should be present in all languages. Three broad groupings of categories that are salient to this thesis are the division between lexical and functional categories, the division between projecting and non-projecting categories, and the division between endocentric and exocentric categories.

Lexical categories are those categories that contribute semantic information to an utterance: in English these are Adv, Adj, N, P, and V representing adverbs, adjectives, nouns, prepositions, and verbs respectively. Functional categories described by Bresnan et al. (2016, p. 104) as "specialized subclasses of lexical categories which have a syncategorematic role in the grammar (such as marking subordination, clause type, or finiteness)" contribute information about the structural relations between lexical categories and potentially also semantic information such as tense, aspect, mood, definiteness, although they do not contribute a semantically weighty PRED value. In English, the functional cat-

---

[4] See Bresnan et al. (2016) for a fuller discussion of the properties of categories.

egories are C, D, and I, representing complementisers, determiners, and tensed auxiliary verbs respectively.

The divisions between projecting and non-projecting categories, and between endocentric and exocentric categories relate to X-bar theory (Chomsky, 1968; Jackendoff, 1977). LFG follows X-bar theory in assuming that a terminal node may project a node above it within its phrase, and that this process can repeat until a maximum number of projections from the head is reached. LFG standardly assumes two levels of phrasal projection, which for a head of category X are termed X′ and XP. Where a head projects phrasal structure and this phrase is dominated by another phrase, the daughter phrase attaches from its maximal projection. For the nodes within a phrase, specific constraints apply to the attachments that may be made at that node: attachments at X′ that are sisters of X are termed *complements*, and attachments at XP that are sisters of X′ are termed *specifiers*. Non-projecting categories are those categories for which only the head appears in c-structure (Toivonen, 2003). The exocentric category S is a phrasal category that is not projected by a head and so does not appear as a terminal node in c-structure.

The labelling function λ maps a node onto a category label. The set of category labels L comprises the available levels of projection for each word category in the language's category set, together with any endocentric categories that the language allows. Thus for a language where X stands for projecting categories, W stands for non-projecting categories, and the exocentric category S is available, the category set is given in (37).

(37) L: {X, X′, XP, W, S} with X and W instantiated for all relevant categories

Various proposals have been made regarding a feature set that might be able to generate the label set L in a principled way. I follow Dalrymple (2017) in assuming that the set of categorial features matches the set of lexical categories, which for English gives the feature set in (38).

(38) [ADJ±, ADV±, N±, P±, V±]

I then follow Grimshaw (2000) in assuming that functional categories take on the categorial features of their lexical complement[5]. This means that for English, the categorial feature specifications for categories C, I, and V are all as in (39).

(39) [ADJ−, ADV−, N−, P−, V+]

3.1.4.1 Universal constraints on c-structure

Besides satisfying the definition given (36), further universal constraints apply to c-structure in LFG. One is a constraint operating on the mother function M and the precedence relationship, specifying that tree branches may not cross (Kaplan, 1995, p. 10).

---

[5] For this thesis I assume that there are additional bar level features and functional features that further distinguish between lexical and functional categories and bar-level categories that otherwise have the same categorial feature specifiation, but I am not making use of this feature set in my subsequent analyses.

Thus the precedence order between two nodes also holds for their mother nodes, specified in (40).

(40) $\mathrm{M}(n_1) \prec \mathrm{M}(n_2) \Rightarrow n_1 \prec n_2$

A second universal c-structure constraint is that one word can map onto only one terminal node, and the ordering of the s-string is preserved in the precedence relationship of the terminal c-structure nodes[6].

A third universal constraint is the principle of economy of expression, which rules out the presence of c-structure nodes that may be licensed by phrase-structure rules but make no additional contribution to a syntactic analysis or the construction of meaning. One definition of the principle of economy of expression is given by Bresnan et al. (2016, p. 90): "All syntactic phrase structure nodes are optional and are not used unless required by independent principles (completeness, coherence, semantic expressivity)."

3.1.4.2 Language-specific constraints on c-structure

Language-specific syntactic constraints apply to c-structure. Besides the language-specific set of categories and their characteristics with regard to projection and endocentricity, each language has its own specific set of phrase-structure rules that govern the attachment possibilities of phrases in the dimensions of immediate dominance (the categories of phrases that are permitted to attach within the dominating phrase) and linear precedence (constraints on the precedence of phrases attaching to a node with respect to each other and with respect to the phrasal head).

The general form of a phrase-structure rule is shown in (41), where A, B, C, and Z are drawn from L, the set of node labels. LFG does not generally restrict the number of daughters of a node, so there is no assumption of binary branching.

(41) $$\mathrm{A} \longrightarrow \underset{\{\mathit{constraints}\}}{\mathrm{B}} \quad \underset{\{\mathit{constraints}\}}{\mathrm{C}} \quad \dots \quad \underset{\{\mathit{constraints}\}}{\mathrm{Z}}$$

A set of c-structure nodes is admissible if, for every node $\bullet_i$ that has the ordered set of daughters $\{\bullet_j, \bullet_{j+1}, \dots \bullet_{j+n}\}$, the relationship (42) is in the set of phrase structure rules for the language (Dalrymple, 2017).

(42) $\lambda(\bullet_i) \longrightarrow \lambda(\bullet_j) \quad \lambda(\bullet_{j+1}) \quad \dots \quad \lambda(\bullet_{j+n})$

The constraints associated with the right-hand categories of the phrase structure rule B, C, …Z in (41) may apply to any level of representation. They are most frequently seen as constraints on f-structure, but are not limited to this level[7]. Right-hand side elements

---

[6] Lowe (2016) follows Wescoat (2002) in proposing that in some constrained cases, a single s-string word may map onto two terminal c-structure nodes. In these cases, each element of the lexical specification may map onto only one of the c-structure nodes, preserving π as a function (p. 167, footnote 16).

[7] The account of nominal adposition in Australian languages proposed by Sadler and Nordlinger (2010) includes meaning constructors that are introduced by phrase-structure rules.

are generally optional unless motivated by the requirement for a lexical head[8] and empty categories are generally not admitted[9].

3.1.5 F-structure

Functional structure in LFG, referred to as f-structure, is the universal, abstract syntactic representation of an utterance. This is encoded in terms of grammatical functions which specify the relationship between an element of the utterance and another element which either governs it, or which it modifies. The element of the sentence which governs, or which is modified by these grammatical functions has a semantic value termed PRED. The grammatical functions in relation to this PRED are drawn from a single set of grammatical functions available to all languages. Dalrymple (2001, p. 9) defines this as shown in (43).

(43) SUBJ, OBJ, OBJ$_\theta$, COMP, XCOMP, OBL$_\theta$, ADJ, XADJ

In (43), θ in OBJ$_\theta$ and OBL$_\theta$ is a variable that is instantiated with a thematic role (Gruber, 1962), e.g. Source, Goal, Location, in an analysis. In this thesis, where the exact role is not pertinent to the discussion, I will refer to OBJ$_\theta$ and OBL$_\theta$.

There are several properties of grammatical functions that can be used to divide the set. For this thesis, the salient properties are governable/non-governable grammatical functions and, within the governable subset, core/non-core grammatical functions. Governable grammatical functions are those that can be selected by a predicate, which include all of the functions in (43) other than the modifier functions ADJ and XADJ. Within the governable grammatical functions, the core functions SUBJ, OBJ, OBJ$_\theta$, and OBL$_\theta$ can be distinguished from the non-core functions COMP and XCOMP. These groupings are illustrated in (44).

(44)

Governable						Non-governable	
Core			Non-core				
SUBJ	OBJ	OBJ$_\theta$	OBL$_\theta$	COMP	XCOMP	ADJ	XADJ

Bresnan and Kanerva (1989) proposed that the grammatical functions SUBJ, OBJ, OBJ$_\theta$, OBL$_\theta$ could be analysed using the features specifications ±R, "thematically restricted", and ±O, "objective", producing the specification matrix in (45).

(45)

	−R	+R
−O	SUBJ	OBL$_\theta$
+O	OBJ	OBJ$_\theta$

Falk (2005) further extended this by adding the features ±C, clausal, and ±S, saturated, and expanding the specification to include all grammatical functions, as shown in (46).

---

[8] Some authors (e.g. Falk, 2001) assume that headless phrases are permitted for a restricted set of categories.

[9] Bresnan et al. (2016) propose a role for empty categories in their account of weak crossover, but more parsimonious accounts (e.g. Dalrymple et al., 2001) that do not assume empty categories are available. In developing an incremental theory, it is more parsimonious not to have to deal with a potential empty category at every point in an utterance.

(46)

		−R	+R	
		+S		−S
−C	−O	SUBJ	OBL$_\theta$	XOBL$_\theta$
	+O	OBJ	OBJ$_\theta$	XOBJ$_\theta$
+C	±O		COMP	XCOMP

Falk's proposed grammatical functions XOBJ$_\theta$ and XOBL$_\theta$ have not been widely adopted, with the function XADJ more generally used in its place. However, I will return to the featural basis of grammatical functions in Chapter 5.

An f-structure is an unordered set of attribute-value pairs $\langle a, v\rangle$, and is thus in itself a function whose domain is the set of attributes and whose range is the set of associated values. Its attributes are grammatical functions and syntactic features. Notwithstanding Bresnan and Kanerva's and Falk's featural analyses of grammatical functions, f-structure attributes are usually assumed to be atomic[10]. Values within f-structure are either atomic (including semantic forms), or are themselves f-structures, or are sets of f-structures. An example f-structure for the sentence *David devoured the cake yesterday* is given at (47).

(47) *David devoured the cake yesterday.*

$$\left[\begin{array}{ll}
\text{PRED} & \text{‘devour}\left\langle \text{SUBJ, OBJ} \right\rangle\text{’} \\
\text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & \text{‘David’} \end{array}\right] \\
\text{OBJ} & \left[\begin{array}{ll} \text{PRED} & \text{‘cake’} \\ \text{DEF} & + \end{array}\right] \\
\text{ADJ} & \left\{ \left[\begin{array}{ll} \text{PRED} & \text{‘yesterday’} \end{array}\right] \right\}
\end{array}\right]$$

An f-structure may be the value of an attribute in more than one other f-structure. This is termed structure-sharing and occurs in cases such as when a discourse function is present or when control is involved in determining a referent. An example is given in (48), where the f-structure g provides the value of the subject of *want*, but also the subject of *eat*, represented by the line joining the two attributes on the diagram. Constraints on structure-sharing are discussed further in Section 3.1.5.6.

(48) *David wants to eat cake.*

$$f\left[\begin{array}{ll}
\text{PRED} & \text{‘want}\left\langle \text{SUBJ, XCOMP} \right\rangle\text{’} \\
\text{SUBJ} & g\left[\begin{array}{ll} \text{PRED} & \text{‘David’} \end{array}\right] \\
\text{XCOMP} & h\left[\begin{array}{ll} \text{PRED} & \text{‘eat}\left\langle \text{SUBJ, OBJ} \right\rangle\text{’} \\ \text{SUBJ} & \\ \text{OBJ} & i\left[\begin{array}{ll} \text{PRED} & \text{‘cake’} \end{array}\right] \end{array}\right]
\end{array}\right]$$

---

[10] Butt (1995) proposes that attributes are themselves composed of a set of feature-value pairs but questioning the standard assumption of LFG is not necessary for this thesis.

3.1.5.1 Projecting f-structure from c-structure: the ϕ-function

Each node in a c-structure projects a single f-structure: the nodes of the overall c-structure form the domain of the ϕ-function whose range is the overall f-structure. A functional constraint annotation on a c-structure node constrains the relationship of the projected f-structure to other f-structures. Example (50) illustrates this for the sentence *David will bake a cake*, using the phrase-structure rules in (49). The subscripts on the NP and N nodes are for reference in the text and have no meaning in the analysis.

(49) a. $$\text{IP} \longrightarrow \underset{(\uparrow\ \textsc{subj}) = \downarrow}{\text{NP}} \qquad \underset{\uparrow=\downarrow}{\text{I}'}$$

b. $$\text{I}' \longrightarrow \underset{\uparrow=\downarrow}{\text{I}} \qquad \underset{\uparrow=\downarrow}{\text{VP}}$$

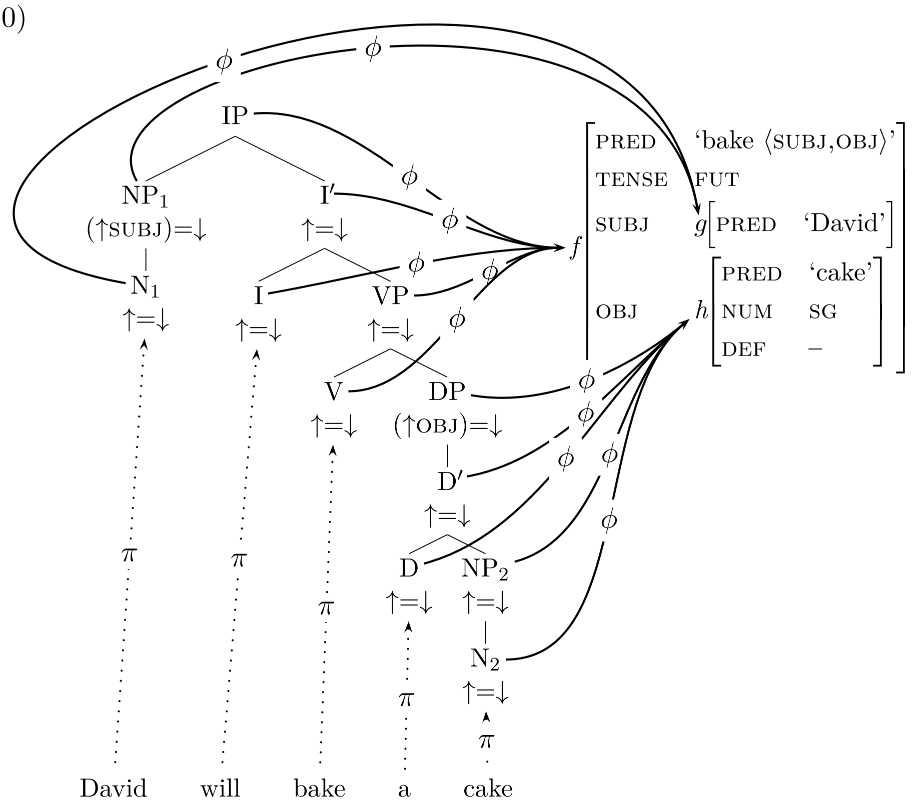
c. $$\text{DP} \longrightarrow \underset{(\uparrow\ \textsc{poss}) = \downarrow}{\text{DP}} \qquad \underset{\uparrow=\downarrow}{\text{D}'}$$

d. $$\text{D}' \longrightarrow \underset{\uparrow=\downarrow}{\text{D}} \qquad \underset{\uparrow=\downarrow}{\text{NP}}$$

e. $$\text{NP} \longrightarrow \underset{\uparrow=\downarrow}{\text{N}}$$

f. $$\text{VP} \longrightarrow \underset{\uparrow=\downarrow}{\text{V}} \qquad \underset{(\uparrow\ \textsc{obj}) = \downarrow}{\text{DP}}$$

(50)



Where a node is annotated with the functional constraint ↑=↓ its f-structure is the same as the f-structure projected by its mother: by convention ↓ represents the f-structure of the current node and ↑ represents the f-structure of the current node's mother[11]. This is always the case for the nodes projected between the head and the maximal projection according to X-bar theory and may also be the case for other nodes, dependent on the phrase-structure rules. Thus for the tree in Figure 50, the nodes IP, I′, I, VP, and V all project f-structure *f*, nodes N_1 and NP_1 both project f-structure *g*, and nodes DP, D′, D, NP_2, and N_2 all project f-structure *h*. Phrase-structure rule (49a) associates the constraint (↑ SUBJ) = ↓ with node NP_1 and hence the semantic information from nodes NP_1 and N_1 is projected into f-structure *g*, similarly rule (49f) associates the constraint (↑ OBJ) = ↓ with the DP node and hence on f-structure *h*.

Functional constraints on the relationship between f-structures are not only introduced by phrase-structure rules, but may also be introduced by the f-structure content of lexical specifications (Nordlinger, 1998). Thus in the Korean sentence in (51), the lexical specification for *pika* 'rain.SUBJ', part of which is given in (52), contributes the inside-out functional constraint (SUBJ ↑), which means "I provide the SUBJ grammatical function to the f-structure projected from my mother node".

(51) *pi-ka* *o-ayo*
rain-SUBJ come-PRS.POL
"It is raining."

(52) *pika* N (↑ PRED) = 'rain'
(SUBJ ↑)

Functional constraints in LFG can also take the form of constraining equations, which use the notation $=_c$. An example of a lexical specification using a constraining equation is given in (53)[12].

(53) *rely* V (↑ PRED) = 'rely ⟨SUBJ,OBJ$_\theta$⟩'
(↑ OBL$_\theta$PRED) $=_c$ 'on⟨OBJ⟩'

Whereas the functional constraint in the first line adds information to f-structure, the constraining equations add a further constraint on the well-formedness of the f-structure without requiring *rely on* to be treated as a single lexical item. Thus the third line of the specification can be read "the PRED value of the OBJ$_\theta$ attribute of my f-structure is only grammatical if it is 'on⟨OBJ⟩'." This accounts for the ungrammaticality of the sentence * *We rely about David* compared to *We rely on David.*

3.1.5.2 Constraints on f-structure well-formedness

An f-structure is grammatical if it satisfies the universal principles of completeness, coherence, and consistency, defined as follows (Kaplan and Bresnan, 1982).

---

[11] ↑ represents the f-structure presented by the current node's mother. If the current node is $*$, the mother node itself can be represented by $\hat{*}$ and M($*$).

[12] My thanks to John Lowe for permission to use an example from his lecture notes.

Completeness: An f-structure is locally complete if and only if all of the grammatical functions selected by its PRED value are present. An f-structure is complete if and only if it and each of the f-structures that it contains are locally complete.

Coherence: An f-structure is locally coherent if and only if all the governable grammatical functions that it contains are governed by a local predicate. An f-structure is coherent if and only if it and all its subsidiary f-structures are locally coherent.

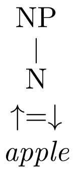
Bresnan and Mchombo (1987, p. 746) extend the coherence condition to include the non-governable grammatical functions as follows. “The extended coherence condition requires that all functions in f-structure be BOUND. An argument function (i.e. a subcategorizable function like SUBJ, OBJ, OBL) is bound if it is the argument of a predicator (PRED). An adjunct is bound if it occurs in a f[unctional]-structure which contains a PRED. Finally, a topic or focus is bound whenever it is functionally identified with, or anaphorically binds, a bound function.”

Consistency: An f-structure is consistent if there is only one value — either an atom or an f-structure, or a set of atoms or f-structures — for each of its attributes.

3.1.5.3 Universal constraints on f-structure

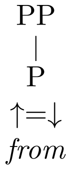
The universal constraints on f-structure discussed in Section 3.1.5.2 apply to all f-structures. Analyses are given below for for three one-word fragments from English (54a), (54b) and Korean (54c), using the phrase-structure rules from Appendix A[13].

(54) a. *apple* N (↑ PRED) =‘apple’



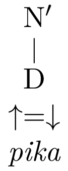
$$\begin{bmatrix}\text{PRED} & \text{‘apple’}\end{bmatrix}$$

b. *from* P (↑ PRED) =‘from ⟨OBJ⟩’



$$\begin{bmatrix}\text{PRED} & \text{‘from}\left\langle\text{OBJ}\right\rangle\text{’}\\ \boxed{\text{OBJ}} & \end{bmatrix}$$

c. *pika* N (↑ PRED) =‘rain’
(SUBJ ↑)



$$\begin{bmatrix}\boxed{\text{PRED}} & \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{‘rain’}\end{bmatrix}\end{bmatrix}$$

The f-structure in (54a) is complete. However, the f-structure in (54b) is incomplete because the required OBJ grammatical function is not present, and the f-structure in (54c) is incoherent because the SUBJ grammatical function contributed by the lexical

---

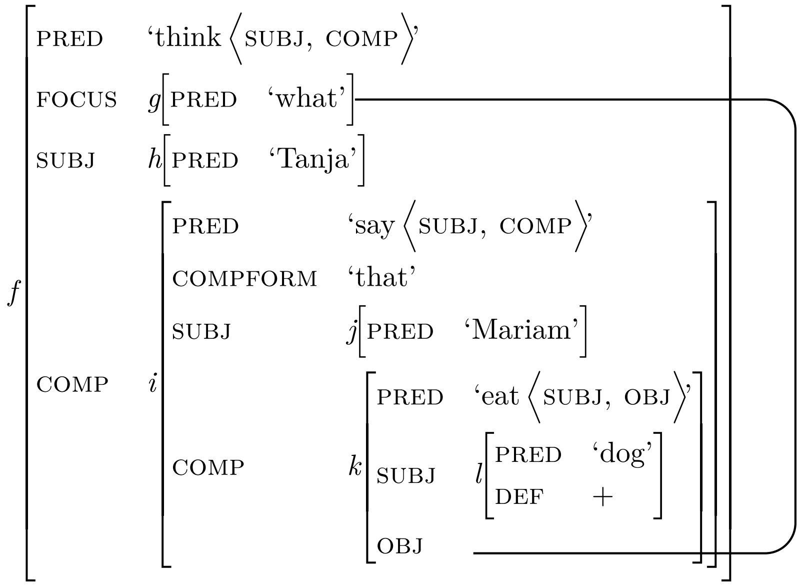
[13] Falk (2001) assumes that lexical categories in English do not project an X′ node. Conversely, Cho and Sells (1995) assume that the maximal projection of lexical cateogries in Korean is X′.

specification for *pi-ka* 'rain-SUBJ' is not governed by a PRED value. In the examples, and in subsequent examples, attributes where a value is required in order to satisfy the principles of completeness and coherence are outlined, e.g. $\boxed{\text{OBJ}}$.

3.1.5.4 Relationships between f-structures

Relations between elements of f-structure can be described in various ways, which I will illustrate with reference to the simplified f-structure in (55), which ignores tense and aspect features.

(55) *What did Tanja think that Mariam said the dog had eaten?*



Containment: Formally, an f-structure is a set of attribute-value pairs. Intuitively, f-structures *g, h, i, j, k*, and *l* are within *f* in (55), but they are not formally members of *f*. Instead, the relationship is described as containment, so *f* contains *g, h, i, j, k*. The containment relation can be specified recursively as in (56).

(56) f-structure f contains f-structure g if either:

a. g is the value of an attribute in f, or

b. h is the value of an attribute in f and h contains g.

F-structure paths: Formally an f-structure is a function from attributes to values. Function composition therefore allows a contained f-structure at any depth of containment to be described in terms of a containing f-structure. For example, the f-structure *j* in (55) is equivalent to (*f* COMP SUBJ). The sequence COMP SUBJ is termed the *path* from *f* to *j*. Paths may be underspecified by using variables in place of specific grammatical functions, or by using a Kleene star or Kleene plus to indicate a variable path length. Thus the containment relationship can also be expressed as in (57), where GF is a variable that can be instantiated by any of the grammatical functions in the universal set, and each element of the path created when the $\textsc{gf}^+$ is expanded is instantiated separately.

(57) $f\ \textsc{gf}^+ = g$

F-structure paths are used in accounts of constraints on long-distance dependencies and will be discussed further in Section 3.1.5.6.

F-command: F-command is a relationship between two f-structures: I assume the definition in terms of inside-out functional uncertainty given by Dalrymple (2001, p. 159), shown in (58). F-structure *f* f-commands f-structure *g* if *f* is not *g*, *f* does not contain *g* and there is another f-structure *h* such that *f* is a value of an attribute of *h*, and *h* contains *g*.

(58) *f* f-commands *g* if:

a $f \neq g \wedge$

b. $f\ \textsc{gf}^+ \neq g \wedge$

c. $\exists h: \begin{array}{lcl} h\ \textsc{gf} & = & f \wedge \\ h\ \textsc{gf}^+ & = & g \end{array}$

Thus in (55), f-structure *g* f-commands *h, i, j, k,* and *l*, but is itself f-commanded only by *h*, *i*, and *l*.

3.1.5.5 Properties deriving from the c-/f-structure interface

There are some properties of f-structure that are dependent on the projection from c-structure: f-precedence and the CAT predicate.

F-precedence: F-precedence relates the contents of functional structure to the linear ordering constraints of c-structure. Bresnan et al. (2016, p. 213) give the definition in (59).

(59) Given a correspondence mapping ϕ between a c-structure and its f-structure, and given two subsidiary f-structures α and β, α f-precedes β if and only if the rightmost node in $\phi^{-1}(\alpha)$ precedes the rightmost node of $\phi^{-1}(\beta)$.

The CAT predicate: The inverse relation of the ϕ function, ϕ^{-1}, is used in the definition of the CAT predicate, which states the relationship between an f-structure and the category labels of its corresponding c-structure nodes. Dalrymple (2017, p. 39) gives the definition of CAT in (60).

(60) Definition of CAT (Crouch et al., 2008; Kaplan and Maxwell, 1996)
$\text{CAT}(f, C)$ iff $\exists n \in \phi^{-1}(f) : \lambda(n) \in C$.
"$\text{CAT}(f, C)$ is true if and only if there is some node n that corresponds to f via the inverse ϕ correspondence (ϕ^{-1}) whose label (λ) is in the set of categories C."

The CAT predicate is used when category constraints apply to the value of grammatical functions at f-structure, such as in unlike-category coordination (Dalrymple, 2017), anaphoric control (Bresnan et al., 2016). Another approach might be to include category information in an f-structure, although this would conflict with the theoretical separation between levels of representation. As noted by Kaplan and Maxwell (1996, p. 93):

> “The CAT predicate permits constraints to be imposed on the categories of the nodes that map to an f-structure without actually copying a category feature into the f-structure (or doing it surreptitiously as Kaplan and Bresnan (1982) did when they used grammatical-function names like VCOMP, ACOMP, and PCOMP). Asserting constraints by means of this correspondence-based predicate maintains the essential modularity of c-structure and f-structure properties while still allowing for limited and controlled interactions to be expressed.”

3.1.5.6 Constraints on structure-sharing

Structure sharing in LFG describes cases where an f-structure is the value of more than one grammatical function in an utterance. This may result from syntactic control of an embedded clause, or from an *unbounded dependency*, where the shared grammatical functions may be separated by considerable material in the s-string. Phenomena covered by the term ‘unbounded dependency’ are often described as fronting, filler-gap dependencies, and/or wh-dependencies. For an utterance to be grammatical, unbounded dependencies must be resolved. This means that it is possible to write an equation whose left and right hand sides are the paths to the two f-structures. In f-structure diagrams, this equivalence has been indicated by a line between the two functions.

Resolution of unbounded dependencies is governed by constraints on the path through f-structure between the shared functions. Because the path may be of an arbitrary length, regular expressions are used in the constraints, including the Kleene star and Kleene plus operators. The term *functional uncertainty* is used to express the fact that, although the overall path between the shared functions is constrained, the exact grammatical functions that are shared are uncertain until the dependency is resolved.

The path between the shared functions can be described in terms of the grammatical function that is less embedded within the outermost f-structure, described as outside-in functional uncertainty, or it can be described in terms of the grammatical function that is more embedded within the outermost f-structure, described as inside-out functional uncertainty. Examples of both, taken from Falk (2001, p. 152) are given at (61).

In the equations, the f-structure represented by (↑GF) is the value of the GF attribute for this f-structure, i.e. it is inside the current f-structure. The f-structure represented by (GF↑) is f-structure for which this f-structure provides the discourse function, i.e. it is outside the current f-structure.

(61) a. Outside-in $(\uparrow \textsc{df}) = (\uparrow \textsc{comp}^{*}\ \textsc{gf})$

b. Inside-out $(\uparrow \textsc{gf}) = ((\textsc{comp}^{*} \uparrow)\ \textsc{df})$

The outside-in equation means that “the discourse function of this f-structure is equal either to another grammatical function of this f-structure, or to a grammatical function of an f-structure whose path from this f-structure consists of any number of COMP grammatical functions.

The inside-out equation means that “the grammatical function GF of this f-structure is equal either to the discourse function of this f-structure, or to the discourse function of an f-structure whose path to this f-structure consists of any number of COMP grammatical functions.

The implications of incrementality for resolving unbounded dependencies are discussed further in Section 3.3.5.

3.1.5.7 C-structure dominance and f-structure containment

Usually, where there is a dominance relation between two c-structure nodes, the f-structures projected by the two nodes are either identical, or the f-structure of the dominating node contains the f-structure of the node it dominates. It is possible to imagine functional constraints on c-structure nodes, or functional constraints introduced lexically, which would introduce information or place constraints on fully specified, non-local f-structures, but for the purposes of this thesis I assume that functional constraints in human languages are restricted either to locally-specified or long-distance functionally-uncertain pathways.

3.1.6 S-structure and its mapping from f-structure

Meaning in LFG is represented at s-structure, which is projected from f-structure by the σ-function. I follow Asudeh and Giorgolo (2012) in treating an s-structure as a function which describes the relationship between an event and its participants. This function is presented as an attribute-value matrix. The domain of all s-structures includes the attributes REL, which specifies the nature of the relationship between the participants, and EVENT, which describes the nature of the event. Lexically-specified, uniquely-indexed ARG attributes map onto the participants in the event, and correspond to the ARG elements of the meaning constructor. Thus the contents of the s-structure allow the propositional content of the utterance to be derived as a proof in Glue Logic (Dalrymple, 1999).

(62)
$$e\left[\begin{array}{ll} \textsc{rel} & \text{bake} \\ \text{EVENT} & ev\left[\quad\right] \\ \textsc{arg}_1 & s\left[\quad\right] \\ \textsc{arg}_2 & c\left[\quad\right] \end{array}\right]$$

I follow Findlay (2016) in assuming the index numbering of ARG attributes represents a general association with a thematic role, rather than solely ensuring uniqueness, and that the mapping function σ between f-structure and s-structure is given by lexically-specified templates that constrain the relationship between grammatical functions and argument positions. The lexical template for a verb sets out the relationship between the verb's core arguments and the argument positions in a universal valency frame, which constrains the grammatical functions that may be associated with a particular argument position. Findlay's proposed universal frame has four possible core argument positions, ARG_1–ARG_4.

Each argument position is associated with a restriction on the grammatical functions with which it can be associated. There is an ordering constraint on associating grammatical functions with argument positions: starting with the least marked grammatical function and proceeding to the most marked, each GF is associated with the highest-ranking open argument position that conforms with the $\pm\text{O}, \pm\text{R}$ restrictions associated with grammatical functions as described in Section 3.1.5.

Further constraints may be added by other lexical items that introduce templates associated with valency-changing operations. These include operations where the grammatical function-thematic role mapping of core arguments changes, such as passivisation shown in (63), which are described by Findlay as *morphosyntactic*, and operations where additional arguments are introduced to the verb alongside a change in core argument mapping, such as the benefactive shown in (64) described by Findlay as *morphosemantic*.

(63) Morphosyntactic passivisation
    a. David baked a cake.
    b. A cake was baked (by David).

(64) Morphosemantic benefactive
    a. David baked a cake.
    b. David baked Stephen a cake.

3.1.7 Meaning and discourse

Many authors working in LFG assume the level i-structure, projected from s-structure, to represent information structure, a term introduced by Halliday (1967) to capture "the status of the elements [of a sentence] ...as components of a message." However, LFG has no formal projection of intersentential discourse structure[14], but because propositional content is computed from meaning constructors whose composition is constrained at s-stucture, this can be fed into another theory of discourse representation. The Partial Compositional Discourse Representational Theory proposed by Haug (2014) on the basis of Compositional Discourse Representation Theory (Muskens, 1996) explicitly considers the

---

[14] King and Zaenen (2004) discusses the differences between the information structure and discourse structure in LFG.

role of pragmatic inference in relation to the incremental increase in available information as utterances unfold.

In Chapter 5, I discuss the relationship between processing and incremental theories, including the possibility that referents may appear in a mental discourse representation without needing explicit computation of the projection relation π and the projection functions ϕ, and σ.

3.1.8 Anaphors

Anaphors may take antecedents that are within, or outside, an utterance. Where syntactic constraints operate within an utterance, LFG predominantly expresses syntactic constraints on the binding of an anaphor to an antecedent in terms of binding domains defined using functional structure, following Dalrymple (1993). The constraint that applies to an anaphor is lexically specialised, and different languages may vary in the binding domains that they use[15]. Binding domains can be described using f-structure path constraints, similarly to the constraints on the resolution of unbounded dependencies. Table 3.1 includes the definition of the universally available binding domains provided by Dalrymple (2001, pp. 283-284).

Table 3.1: Universally available binding domains (Dalrymple, 2001)

Domain	Description	Path specification
Coargument Domain	minimal domain defined by a PRED and the grammatical functions it governs	$\left(\begin{array}{cc} \text{GF}* & \text{GF}_{\text{pro}}\ f \\ \neg(\rightarrow \text{PRED}) & \end{array}\right)$
Minimal Complete Nucleus	minimal domain with a SUBJ function	$\left(\begin{array}{cc} \text{GF}^* & \text{GF}_{\text{pro}}\ f \\ \neg(\rightarrow \text{SUBJ}) & \end{array}\right)$
Minimal Finite Domain	minimal domain with a TENSE feature	$\left(\begin{array}{cc} \text{GF}^* & \text{GF}_{\text{pro}}\ f \\ \neg(\rightarrow \text{TENSE}) & \end{array}\right)$
Root Domain	f-structure of the entire utterance	$(\text{GF}^*\ \text{GF}_{\text{pro}}\ f)$

Binding constraints may be positive, whereby the anaphor and its antecedent must both be in the same binding domain, or negative, where the anaphor and its antecedent must not be the same binding domain. An example of a positive binding constraint would be the case of English reflexive pronouns, which must be bound in the Minimal Complete Nucleus by an antecedent that f-commands them. Failure to identify an antecedent within the binding domain results in ungrammaticality. An example of a negative binding constraint would be on the English pronouns *he, him*, which cannot be bound by an antecedent within the Coargument Domain (Dalrymple, 2001, p. 287).

---

[15] Other constraints which can apply include f-precedence constraints, and s-structure constraints expressed in terms of a hierarchy of thematic roles (Bresnan and Kanerva, 1989).

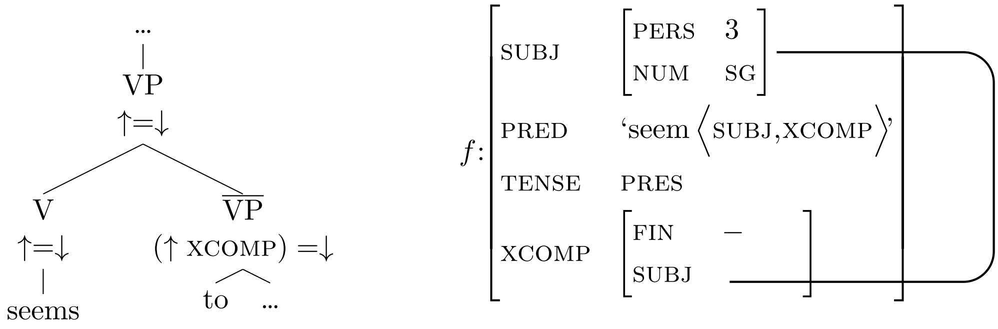
3.2 Fragments and incrementality in LFG

In this section I discuss the ability of LFG to analyse language fragments and introduce the proposal of Asudeh (2012) as the basis of an incremental theory as a series of incrementally growing fragments that begin with the first word of an utterance. From Asudeh's outline, I suggest desiderata for a detailed incremental theory of LFG.

3.2.1 LFG analyses of language fragments

LFG is a declarative framework and there are no assumptions of movement or derivation during an analysis. C-structure rules also assume that a terminal node appears only if generated by the presence of a word. It is therefore possible to generate LFG analyses of incomplete utterances, as demonstrated by Bresnan (2001, pp.88-92) with regard to the sentence fragment in (65).

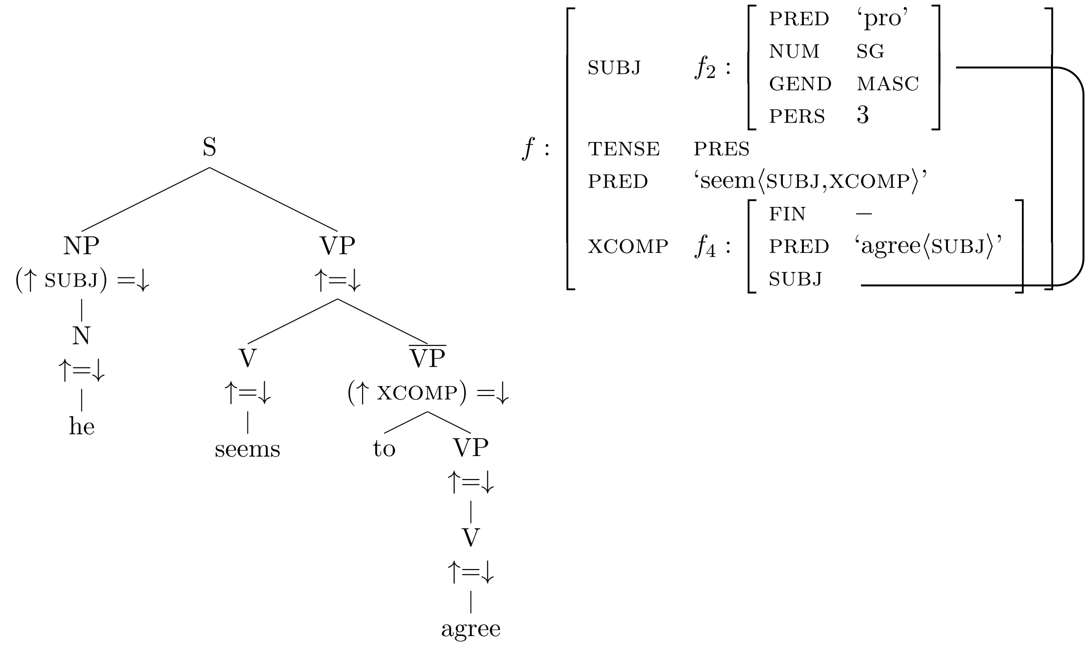
(65) ...seems to...



The analysis in (65) meets neither the conditions of completeness or coherence necessary for an utterance to be grammatical (Dalrymple, 2001). However, it is possible to formally represent not only the partial c-structure and f-structure given above, but also the semantic content of a fragment and its contribution to a discourse. To this end, Bresnan proceeds to demonstrate how, in a given context (66), syntactic and semantic content can be inferred to produce a complete and coherent analysis, in which the f-structure in (65) is subsumed into a fuller f-structure in (67).

(66) [Speaker A:] He agrees?
[Speaker B:] — seems to.

(67) He seems to agree.



Thus LFG's modular architecture allows a unified grammatical treatment of a range of processing tasks including syntax and semantics. As a declarative framework, there are no additional assumptions needed to generate real-time analyses during an utterance, and so the framework is equally applicable to production and comprehension without needing to make a claim that the two processes are identical. I now turn specifically to incremental analyses and some of the theoretical challenges that remain to be addressed.

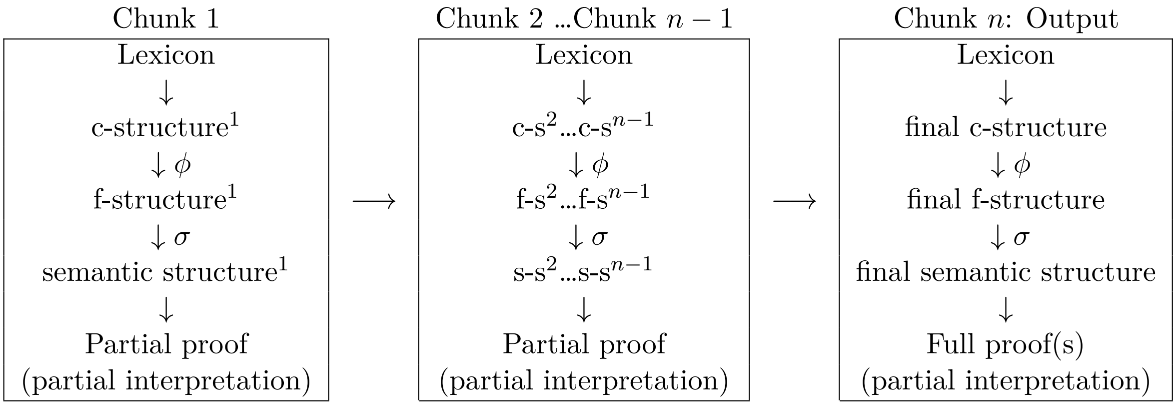
3.2.2 Explicit approaches to incrementality in LFG

Incrementality in LFG has been addressed by Asudeh (2012), who considers resumptive pronouns in English[16]. He develops a processing account that shows how a complex sentence that is ungrammatical overall can be composed of locally well-formed elements. As such, he assumes the building of partial f-structures in incremental steps, illustrated in Figure 3.2. However, this process is not described in detail, nor is there exposition of the partial c- and s-structures that coexist with the partial f-structures. Also, no formal relationship has been defined between any structure generated at one chunk and its counterpart generated at the subsequent chunk.

---

[16] Block and Hunze (1986) describe the incremental construction of c- and f-structure during parsing, but this is based on the Earley algorithm (1970) rather than an elaboration of an incremental processing theory.

Figure 3.2: Asudeh's model (2012, p.301) for parsing and interpretation



Besides showing how an incremental model of LFG is structured, Asudeh (2012, p.301) provides the following general assumptions (68) for a model of processing:

(68) a. Production and parsing are incremental.

b. Incremental production and parsing attempt to construct locally well-formed structures.

c. Global well-formedness applies only to the output of production and parsing.

d. Production and parsing are constrained by memory limitations based on complexity factors, including distance, structural complexity, and intersecting interpretations of unbounded dependencies.

3.2.3 The aims of an incremental theory of LFG

In developing an incremental theory of LFG I am seeking to address the questions raised by Asudeh's assumption (68b), that incremental production and parsing are attempting to construct locally well-formed structures. This is distinct from a theory of grammar in processing, which must also address the cognitive constraints inherent in Asudeh's assumption (68d). The aims that I set out below are for an incremental theory based on a grammar whose rules and constraints provide an account of global well-formedness, and are described in relation to this grammar.

(i) The theory should include all aspects of the contribution of language form to the available meaning.

As such, an incremental theory needs not only to build syntactic structure, but also to account for the development of semantic content, information structure, and discourse representation. Although it is outside the scope of this project, an incremental theory should be able to include the contribution of phonological representations above the level of the word (e.g. prosody, syntactic mutations in Welsh).

(ii) The theory should be explicit about the calculations that are taking place within an analysis at each word.

The language input will provide different amounts of information to each element of the form-meaning correspondence. For example, a complementiser constrains syntactic structure but does not introduce a new discourse referent or event. The degree of specification and underspecification should be describable in detail for each element, at each word.

LFG c-structure includes constraints on word order and constituent hierarchy, and so emerging structure needs to be mappable onto a c-structure that conforms to these constraints.

Similarly, an f-structure representation of a well-formed sentence must satisfy the principles of completeness, coherence, and consistency, as well as the uniqueness constraint on PRED values. Again, incremental f-structure representations must satisfy these constraints.

Functional constraints apply to unbounded dependencies and anaphora resolution. As unbounded dependencies and anaphora are encountered during an unfolding sentence, the account should identify to what extent the relevant functional constraints are calculable at that point, and the impact this has on potential interpretations.

(iii) The theory should be explicit about the calculations necessary to derive the analysis at any given point from the analysis at the previous word.

This includes calculations within a particular element of the analysis (e.g. the analysis at $word_n$), and also calculations between elements of the analysis (e.g. the relationship between the analyses at $word_n$ and $word_{n+1}$). The complexity of calculation required may not be the same over the whole analysis, and constraints in one element may subsequently have an impact on another element.

(iv) The theory should be able to generate all possible parses of an utterance, and identify the choice points where adding a word results in some previously possible parses no longer being possible.

(v) Where adding a word to an analysis introduces ambiguity by providing a range of possible interpretations, the theory should be able to identify at which elements of the analysis this ambiguity arises, and the contribution of each element to the selection of a particular pathway.

(vi) The theory should specify universal and language-specific effects.

LFG is explicit about which elements of a grammar are universal and which are language-specific, including the constraints on levels of representation discussed in Section 3.1. The incremental theory should ensure that these are identifiable, so that predictions can be made about universal and language-specific grammatical effects on language processing.

(vii) The theory should be specific about the sequencing of computations.

LFG levels of representation are ordered. This allows hypotheses to be tested about the processing time needed to project a level from its predecessor in the series. It also allows a more general hypothesis to be tested, that the ordering of LFG levels of representation corresponds to a sequence of processing tasks.

If these aims are achieved, the theory is in a position to make empirically testable predictions about the processing load during a sentence, not only from structure-building, but also from other linguistic elements. It can also make testable predictions about what types of calculation are *outside* those necessary for an incremental analysis, and thus contribute to a more specific theory of the role of grammar in processing.

3.3 Building syntactic structure incrementally

In Chapter 2, I highlighted how discussions to date on incremental building of language structure usually focus on representations of constituent structure. An LFG analysis of an utterance requires a c-structure in order to project f-structure. However, LFG levels of representation are mutually constraining, and accordingly all levels of representation should be considered when describing the incremental growth of a structural representation of language. In this section I focus on the two syntactic levels of representation, c-structure and f-structure. As discussed in Chapter 2, I assume that incremental parsing entails that lexical information is immediately incorporated into a structural representation: there is no capacity for the information from a word to be held pending further input. I also assume that adding a constituent only adds the minimum additional structure necessary to allow attachment. This reflects Frazier's Minimal Attachment strategy, but in LFG the additional structure may be present in f-structure rather than a constituent structure node following Frazier's definition.

Elements of an analysis that represent the cumulative incremental analysis from the start of the utterance to word n are represented as S_n. Elements that refers to the partial incremental analysis for a single word n are represented as S'_n.

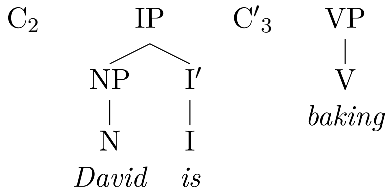
Monotonicity is a fundamental property of a declarative LFG analysis. In the discussion that follows, I treat monotonicity as a constraint and will identify instances where elements of an incremental analysis *do not* increase monotonically through the utterance, i.e. where for sets S_n and S_{n+1}, $S_n \nsubseteq S_{n+1}$.

3.3.1 Incremental growth of c-structure

The well-formedness of a c-structure is determined by its conformity to dominance and precedence constraints. These constraints have universal and language-specific elements:

the language-specific elements are described by phrase-structure rules. C-structure grows when the c-structure C'_{n+1} projected from $word_{n+1}$ in the s-string is combined with the c-structure C_n generated by $word_1 ... word_n$ to produce the c-structure C_{n+1}. Thus in example (69) C_2 represents the c-structure projected from the first two words *David is*, and C'_3 represents the c-structure projected from the third word *baking* before it is attached.

(69)



Each one of these c-structures must be well-formed in terms of their formal elements: the sets N and L, the functions M and λ, and the precedence relation $\prec$. Well-formedness assessment of universal constraints involves the function M and the precedence relation $\prec$. Language-specific constraints are assessed by comparing the labelling function λ with phrase-structure rules.

3.3.1.1 Universal precedence and dominance constraints

I consider first the universal constraints on c-structure which relate to

(i) precedence:

  - the precedence order of terminal c-structure nodes corresponds to the order of the s-string;
  - tree branches may not cross; and

(ii) dominance:

  - each non-root node has only one mother;
  - a c-structure has only one root node.

The s-string is a list (an ordered set) that increases monotonically as an utterance is produced. Because the s-string order is reflected in the precedence relationship between terminal nodes and terminal c-structure nodes are projected by s-forms, the terminal node projected by $word_n$ must be the rightmost terminal node in C_n. Incremental growth of c-structure introduces a new rightmost terminal node, which must be preceded by all of the other terminal nodes. Because of the non-crossing branches constraint (40) introduced earlier, the nodes introduced by C'_{n+1} must be added to the right hand edge of the c-structure.

The right hand edge of a c-structure is defined formally using the function D_r over c-structure nodes (Mycock and Lowe, 2013) which maps a node onto its rightmost daughter (70)[17]. The rightmost daughter of node $*$ is n where n is a member of the set of daughters of $*$ and there is no other daughter of $*$ that is preceded by n.

(70) $D_r(*) \equiv \text{node } n, \text{ where } n \in \mathrm{M}^{-1}(*) \ \wedge \ \neg\exists x.x \in \mathrm{M}^{-1}(*) \ \wedge \ n \prec x.$

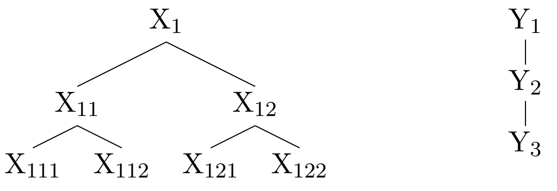
Applying D_r recursively from the root node, the right edge of a c-structure is the set of nodes R that satisfy the conditions in (71). A node $*$ is in the right edge set R of a c-structure if either it has no mother node (and therefore is the root of the c-structure), or if its mother node is in R and $*$ is the rightmost daughter of its mother. As a reminder, $\hat{*}$ denotes the mother of node $*$, equivalent to M($*$).

(71)
$$* \in R \Rightarrow \ \neg\,\hat{*} \vee \left(\hat{*} \in R \ \wedge \ * = D_r(\hat{*}) \right)$$

Once new nodes have been added to the c-structure C_n at its right edge R_n, the resulting c-structure C_{n+1} must also be well-formed: it can have only one root node, each non-root node may have only one mother, and the terminal node of C'_{n+1} must also be the rightmost terminal node of C_{n+1}. The first two conditions require that either the root node of C_n or the root node of C'_{n+1} becomes a daughter node in the new structure, and that all nodes within C'_{n+1} are in the right edge set R_{n+1} of the resulting c-structure. As will be shown in the following examples, there is no relationship between the right edge sets R_n and R_{n+1} and no requirement for monotonic growth of the right edge.

Application of these constraints still admits a number of attachment possibilities: the new nodes may be introduced above the topmost node of the right edge set, at a node within the right edge set, or between two nodes in the right edge set. These possibilities are illustrated below with reference to the c-structures in (72), where the tree of X nodes represents C_n and the tree of Y nodes represents C'_{n+1}. The right edge set R_n is {X$_1$, X$_{12}$, X$_{122}$}. For illustration, I assume that attachment at the terminal nodes X$_{ijk}$ is not possible.

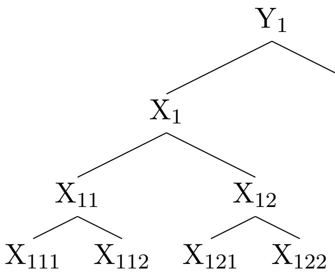
(72)

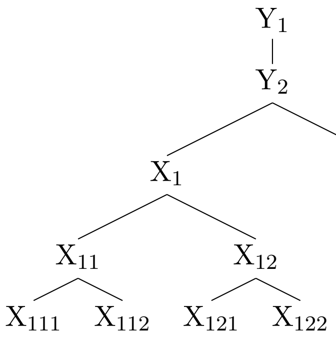


a. Attachment *above* the right edge: the root node X$_1$ of C_n attaches as the leftmost daughter of a member of C'_{n+1}.

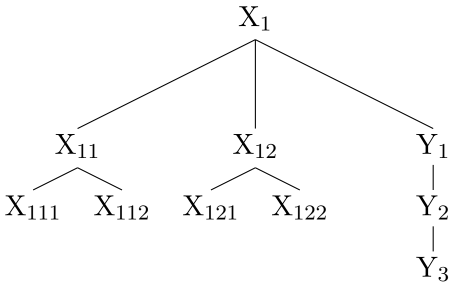
---

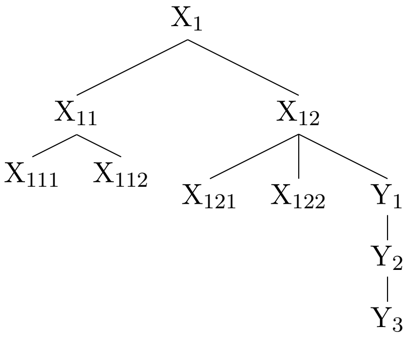
[17] Mycock and Lowe (2013) introduce the relationship $D(*)$ that derives the set of daughters of a node $*$, which is formally equivalent to $\mathrm{M}^{-1}(*)$.



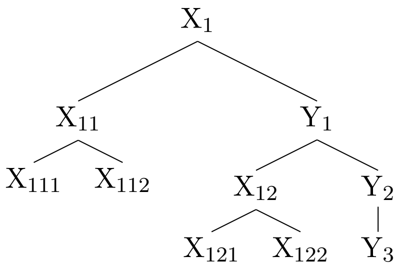


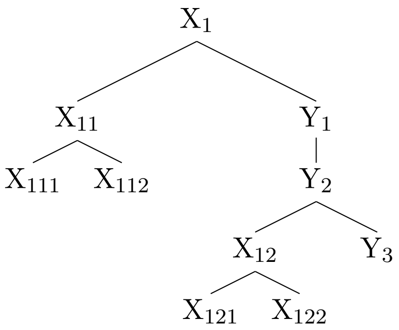
b. Attachment *at* the right edge: the root node of Y_1 of C'_{n+1} attaches as the rightmost daughter of a node n which is member of R_n, with all other daughters of n remaining in the set $M^{-1}(n)$.

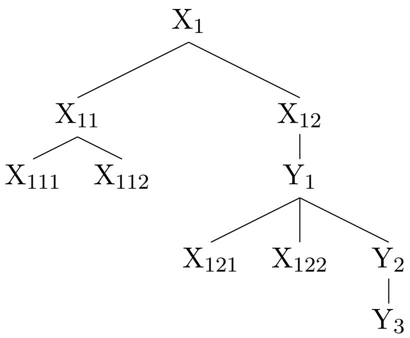


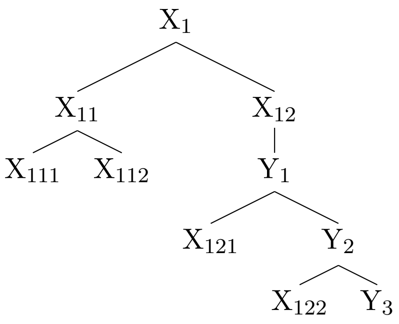


c. Attachment *within* the right edge: the root node Y_1 of C'_{n+1} attaches as the rightmost daughter of a node n which is member of R_n, with at least one other daughter of n becoming a daughter of a node within C'_{n+1}[18].









---

[18] The examples shown here are general, with arbitrary node labels to show how the structures combine. When category labels and phrasal projections are introduced, there is a further constraint on attachment *within* the right edge. This is that the attaching nodes may not interrupt the nodes projected in accordance with X-bar theory. In other words, in the diagrams at (72c) the X nodes that become left daughters of Y nodes must be maximal phrasal projections.

The impact of this is that all of the members of C_{n+1} that were previously members of C_n precede all of the members of C_{n+1} that were previously in C'_{n+1}. Thus $\prec_n \subseteq \prec_{n+1}$, and so the precedence relationship increases monotonically with the s-string.

In (72a) and (72b), the mother function M increases monotonically (more nodes are brought into the domain, and none of the domain-range mappings change), and hence the dominance relationship also increases monotonically. Attachment within the right edge (72c) requires re-mapping of some mother nodes, and so this function does not increase monotonically. However, even when attaching within the right edge, the dominance relationship between nodes is maintained, and so the c-structure dominance relationship increases monotonically regardless of the type of attachment that takes place.

3.3.1.2 Language-specific dominance and precedence constraints

We now have a constrained set of nodes R_n where attachment can take place, together with three types of attachment involving a member of R_n. The possibilities for attachment are further constrained by the language-specific phrase-structure rules, which govern the dominance and precedence relations between categories at different levels of phrasal projection.

We can take phrase-structure rules to have the general form in (73), where $\{\Sigma\mathrm{P}\}$ stands for a maximal phrasal projection from one of a set of categories and the subscripts $\prec X^n$ and $X^n \prec$ mean that the daughter phrase precedes or is preceded by the head X^n respectively.

(73) $\mathrm{X}^{n+1} \rightarrow \{\Sigma\mathrm{P}\}^*_{\prec\mathrm{X}^n}\ \ \mathrm{X}^n\ \ \{\Sigma\mathrm{P}\}^*_{\mathrm{X}^n\prec}$

By considering the whole set of phrase-structure rules for a language, it is possible to generate a daughter set X^d for each head, comprising the categories that may be daughters of a node within the phrasal projection as shown in (74).

(74) $X^d : \{\Sigma_a, \Sigma_b, \Sigma_c, \Sigma_d, \ldots\}$

Two subsets of the daughter set can be distinguished, $X^d_{d\prec\mathrm{X}}$, the set of categories that may attach as a daughter preceding the head, and $X^d_{\mathrm{X}\prec d}$ the set of categories that may attach as a daughter following the head. Depending on the ordering constraints of the language, a particular category may appear in both $X^d_{d\prec\mathrm{X}}$ and $X^d_{\mathrm{X}\prec d}$.

Corresponding mother sets X^m can also be generated for each head, comprising the categories whose phrasal projections may immediately dominate a head's maximal projection (75).

(75) $X^m : \{\Sigma_a, \Sigma_b, \Sigma_c, \Sigma_d, \ldots\}$

Again, two subsets of the mother set can be distinguished: $X^m_{\mathrm{X}\prec m}$ where the attaching phrase precedes the head of the mother phrase, and $X^m_{m\prec\mathrm{X}}$ where the head of the mother phrase precedes the attaching phrase.

In general, for a phrase XP to attach as a daughter of a phrase projected by Y (i.e. at Y′ or YP), the condition in (76) must apply[19].

(76) $\mathrm{X} \in Y^d \wedge \mathrm{Y} \in X^m$

For a phrase projected by a head of category X to attach as a pre-head daughter of a phrase projected by a head of category Y, this can be framed more specifically as example (77).

(77) $\mathrm{X} \in Y^d_{d \prec \mathrm{Y}} \;\wedge\; \mathrm{Y} \in X^m_{\mathrm{X} \prec m}$

Similarly for a Y-headed phrase to attach as a post-head daughter within a X-headed phrase, the more-specific constraint is shown in (78).

(78) $\mathrm{Y} \in X^d_{\mathrm{X} \prec d} \;\wedge\; \mathrm{X} \in Y^m_{m \prec \mathrm{Y}}$

In (72) the three types of attachment were identified. For the first possibility (72a), attachment above the right edge set, where C_n whose root node is a projection of X attaches as a pre-head daughter within C'_{n+1} projected by head Y, we know the category of the heads of both attachment sites and so (77) fully states the constraint.

The second possibility (72b) entails the phrase projected by $word_{n+1}$ attaching as the rightmost daughter of a node that is a member of R_n, the set of nodes along the right edge of C_n. Right attachment of an Y-headed phrase at a node n within set R_n is possible if n is a projection of a head X that satisfies the criteria in (78). The complete specification of this constraint is shown in (79): the four terms of the constraint are that the node is a member of the right edge set; that the value of the x feature of its labelling function is +, that the Y-headed phrase can attach as a right daughter because category Y is a member of the set of X's permitted post-head daughter categories $X^d_{\mathrm{X} \prec d}$; and that the XP can have a right daughter of category Y because the category X is a member of the set of Y's permitted pre-head mother categories $Y^m_{m \prec \mathrm{Y}}$.

(79) $$\begin{aligned} &\exists n . n \in R \\ &\quad \wedge\; (\lambda(n)\ \textsc{x}) = + \\ &\quad \wedge\; \mathrm{Y} \in X^d_{\mathrm{X} \prec d} \\ &\quad \wedge\; \mathrm{X} \in Y^m_{m \prec \mathrm{Y}} \end{aligned}$$

The third possibility (72c) further constrains (79) by requiring that a node within a YP is a legitimate mother node of a projection of X. The specification of this more restrictive constraint is shown in (80): the first four terms of the constraint are analogous to those previously described for (79); the fifth term adds the condition that a YP can have a left daughter of category X because Y appears in the X's set of permitted post-head mothers $X^m_{\mathrm{X} \prec m}$.

---

[19] Some grammars, including Falk (2001), admit headless phrases for particular categories. The conditions set out here apply whether or not a phrasal head is present. However, the presence of absence of a head has other implications for incremental growth, which are discussed further in Section 3.3.1.3.

$$(80)\quad \exists n.n \in R \\ \wedge\ (\lambda(n)\ \mathrm{x}) = + \\ \wedge\ \mathrm{Y} \in X^{d}_{\mathrm{X}\prec d} \\ \wedge\ \mathrm{X} \in Y^{m}_{m\prec \mathrm{Y}} \\ \wedge\ \mathrm{Y} \in X^{m}_{\mathrm{X}\prec m}$$

Having described the constraints in abstract form, I will give some examples from English, using the phrase structure rules from Falk (2001) which are presented in Appendix A. The tables below show the mother-daughter category combinations for which attachment is permitted. Table 3.2 shows the combinations where the daughter is to the left of the head of the mother phrase. These constraints apply where the next word attaches *above* the right edge, or to the existing c-structure below the attachment point when the next word attaches *within* the right edge.

Table 3.2: Daughter precedes head of mother phrase

Daughter phrases: Succeeding mother sets $\mathrm{X}^{m}_{\mathrm{X}\prec m}$: read along rows	Mother node categories Preceding daughter sets $\mathrm{X}^{d}_{d\prec \mathrm{X}}$: read down columns										
	CP	C′	IP	I′	S	DP	D′	AP	NP	PP	VP
CP			*		*						
IP											
S											
DP	*		*		*	*					
AdjP	*		*		*				*		
AdvP	*		*		*			*		*	*
NP	*		*		*						
PP	*		*		*						*
VP	*		*		*						

Table 3.3 shows the combinations where the daughter is to the right of the head of the mother phrase. These constraints apply where the next word attaches *at* the right edge, or to the existing c-structure above the attachment point when the next word attaches *within* the right edge.

Table 3.3: Head of mother phrase precedes daughter

Mother node categories Succeeding daughter sets $X^{d}_{X \prec d}$: read down columns											Daughter phrases: Preceding mother sets $X^{m}_{m \prec X}$: read along rows
CP	C′	IP	I′	S	DP	D′	AP	NP	PP	VP	
							∗	∗		∗	CP
	∗						∗	∗	∗	∗	IP
	∗						∗	∗	∗	∗	S
									∗	∗	DP
										∗	AdjP
									∗	∗	AdvP
						∗			∗	∗	NP
							∗	∗	∗	∗	PP
	∗		∗	∗						∗	VP

Examples of attachment above and attachment at the right edge can be seen by considering the attachment of the second word *is* and the sixth word from in *This should keep the wolf from the door*[20]. To attach *should*, the two c-structures are shown in (81).

(81)

All of the nodes in C$_1$ are in the right edge set R_1 and so the only relevant head category to consider is D. C′$_2$ is a projection of a head of category I. From the phrase-structure rules for English in Appendix A, the mother and daughter sets for categories DP, D′, IP, and I′ (82) can be derived.

(82)
a. $\mathrm{DP}^{m}_{\mathrm{D} \prec m}$:{CP, DP, IP, S}
b. $\mathrm{IP}^{d}_{d \prec \mathrm{I}}$:{AdjP, AdvP, CP, DP, NP, PP, VP}
c. $\mathrm{I}'^{d}_{d \prec \mathrm{I}}$: $\emptyset$
d. $\mathrm{DP}^{d}_{\mathrm{D} \prec d}$: $\emptyset$

---

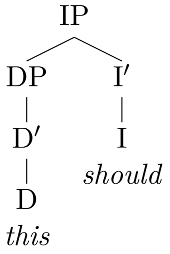
[20] Where there is category ambiguity for a word, I assume for this example that the correct category is selected.

e. $\mathrm{D'}^{d}_{\mathrm{D}\prec d}$:{N}

f. $\mathrm{IP}^{m}_{m\prec \mathrm{I}}$:{AP, CP, NP, PP, VP}

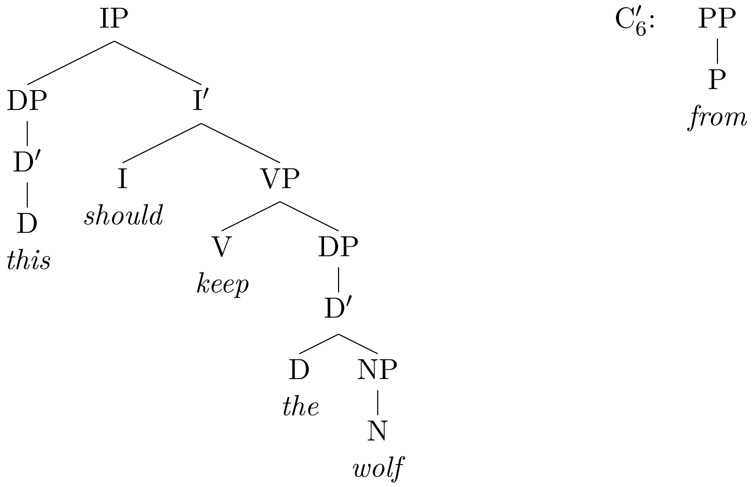
For C'_2 to attach above C_1, either category IP or I′ must appear in (82a) and category D must appear in either (82b) or (82c), corresponding to the projection of I that appears in (82a), and this is the case. Conversely, for C'_2 to attach as right daughter at the right edge of C_1, category IP must appear in either (82d) or (82e) and either category DP or D′ must appear in (82f), but this is not the case. Accordingly attachment takes place *above* the right edge, giving the structure in (83).

(83) C_2:



To attach *from*, the initial c-structures are shown in (84).

(84) C_5:



For left attachment of C_5 into C'_6, the relevant mother and daughter sets are shown in (85).

(85) a. $\mathrm{IP}^{m}_{\mathrm{I}\prec m}$: $\emptyset$

b. $\mathrm{PP}^{d}_{d\prec \mathrm{P}}$: $\{\mathrm{AdvP}\}$

Set (85a) is the empty set: in the grammar in Appendix A, IP cannot attach as the left daughter of a phrase, and so attachment above is not possible.

The right edge set of C_5, R_5 is $\{\mathrm{IP},\mathrm{I}',\mathrm{VP},\mathrm{DP},\mathrm{D}',\mathrm{NP}\}$, and so these are values of the relevant daughter sets where $\mathrm{X}\prec d$, shown together with the mother set $\mathrm{PP}^{m}_{m\prec \mathrm{P}}$ at (86).

(86) a. $\mathrm{DP}^{d}_{\mathrm{D}\prec d}$: $\emptyset$

b. $\mathrm{D}'^{d}_{\mathrm{D}\prec d}$: $\{\mathrm{NP}\}$

c. $\mathrm{IP}^{d}_{\mathrm{I}\prec d}$: $\emptyset$

d. $\mathrm{I}'^{d}_{\mathrm{I}\prec d}$: $\{\mathrm{VP}\}$

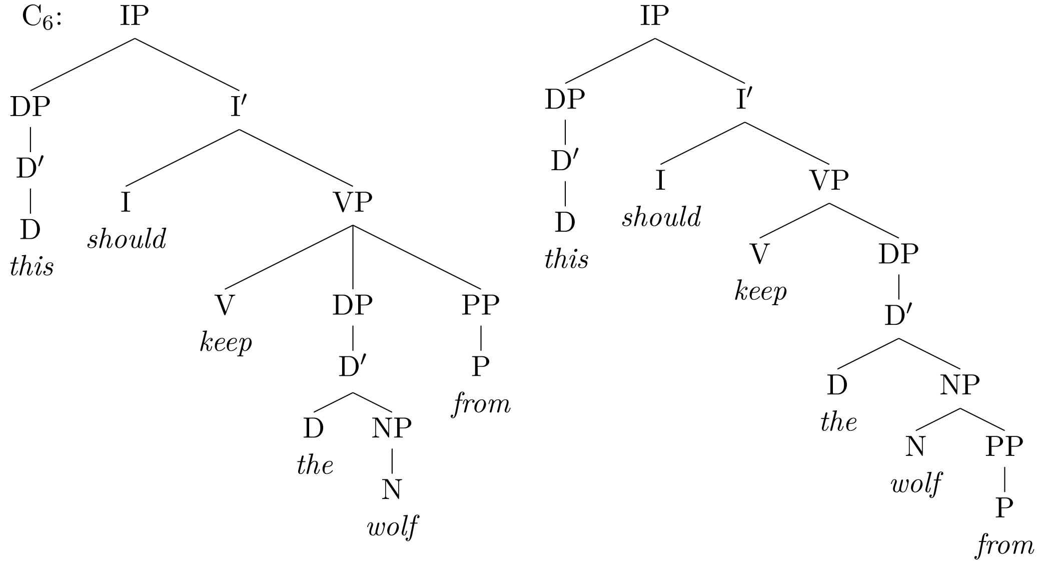
e. $\mathrm{NP}^{d}_{\mathrm{N}\prec d}$: $\{\mathrm{CP}, \mathrm{IP}, \mathrm{PP}, \mathrm{S}\}$

f. $\mathrm{VP}^{d}_{\mathrm{V}\prec d}$: $\{\mathrm{AP}, \mathrm{CP}, \mathrm{DP}, \mathrm{IP}, \mathrm{NP}, \mathrm{PP}, \mathrm{S}\}$

g. $\mathrm{PP}^{m}_{m\prec \mathrm{P}}$: $\{\mathrm{AP}, \mathrm{NP}, \mathrm{PP}, \mathrm{VP}\}$

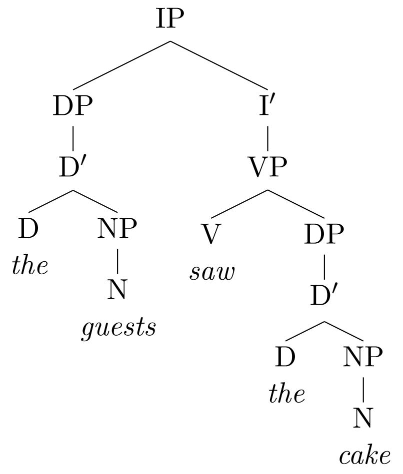
PP appears in sets (86e) and (86f), and NP and VP both appear in (86g). Accordingly, C'_6 can attach at either the VP or the NP resulting in one of the two c-structures in (87).

(87) C_6:

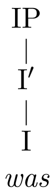


Attachment *within* the right edge was not possible in the previous case because there was no mother-daughter pair in C_5 for which the mother was in the set $\mathrm{PP}^{m}_{m\prec \mathrm{P}}$ and the daughter was in the set $\mathrm{PP}^{d}_{d\prec \mathrm{P}}$. However these conditions are found during incremental growth of sentences in English. One example is when the OBJ of a verb that selects either an OBJ or a COMP argument is reanalysed as the subject of the COMP, as in the processing of *was* in the utterance *The guests saw the cake was still being decorated.* The initial state is shown in (88).

(88) C_5:



C'_6:



The mother and daughter sets for C_5 to attach as a left daughter of C'_6 are shown in (89).

(89) a. $\mathrm{IP}^{m}_{\mathrm{I}\prec m}$: $\emptyset$

b. $\mathrm{IP}^{d}_{d\prec \mathrm{I}}$: {AdjP, AdvP, CP, DP, NP, PP, VP}

c. $\mathrm{I}'^{d}_{d\prec \mathrm{I}}$: $\emptyset$

As in (85a), it is not possible for a c-structure whose root node is IP to attach into another c-structure as a left daughter. The mother and daughter sets for attachment of C'_6 as a right daughter are shown in (90): the first six sets are identical to those in (86).

(90) a. $\mathrm{DP}^{d}_{\mathrm{D}\prec d}$: $\emptyset$

b. $\mathrm{D}'^{d}_{\mathrm{D}\prec d}$: {NP}

c. $\mathrm{IP}^{d}_{\mathrm{I}\prec d}$: $\emptyset$

d. $\mathrm{I}'^{d}_{\mathrm{I}\prec d}$: {VP}

e. $\mathrm{NP}^{d}_{\mathrm{N}\prec d}$: {CP, IP, PP, S}

f. $\mathrm{VP}^{d}_{\mathrm{V}\prec d}$: {AP, CP, DP, IP, NP, PP, S}

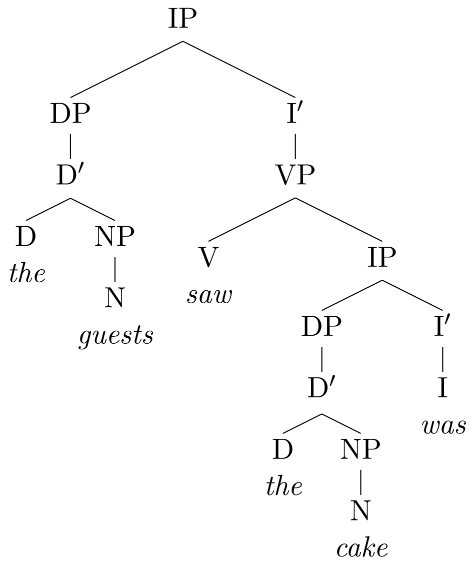
g. $\mathrm{IP}^{m}_{m\prec \mathrm{I}}$:{AdjP, AdvP, C′, NP, PP, VP}

From (90c-e), IP appears in sets $\mathrm{NP}^{d}_{\mathrm{N}\prec d}$ and $\mathrm{VP}^{d}_{\mathrm{V}\prec d}$, and N and V each appear in $\mathrm{IP}^{m}_{m\prec \mathrm{I}}$, so attachment of C'_6 as a right daughter of the VP or NP node in C_5 is possible, similar to the attachment of the PP in the previous example[21]. However, because there is a VP-DP mother daughter pair in C_5 and DP appears in set $\mathrm{IP}^{d}_{d\prec \mathrm{I}}$ (89b), it is also possible for C'_6 to attach within the right edge at this point, producing the c-structure in (91).

---

[21] Functional constraints rule out these possible attachment sites, which will be discussed further in section 3.3.2.

(91) C_6:



3.3.1.3 Introducing underspecified nodes to c-structure

It is possible that the constraints described above prevent a node attaching directly. This is problematic for the theory, because situations like this arise during the incremental growth of grammatical utterances.

Consider an example from English: the attachment of *incredibly* to the c-structure projected by *the*, as part of the grammatical noun phrase *the incredibly inept politician*. The c-structures C_1 and C'_2 are shown in (92).

(92) C_1:

C'_2:

From the phrase-structure rules for English in Appendix A, the mother and daughter sets for categories DP, D′ and AdvP (93) can be derived.

(93) a. $\mathrm{DP}^{m}_{\mathrm{D}\prec m}$:{CP, DP, IP, S}

b. $\mathrm{AdvP}^{d}_{d\prec \mathrm{A}}$:{AdjP, AdvP}

c. $\mathrm{DP}^{d}_{\mathrm{D}\prec d}$: $\emptyset$

d. $\mathrm{D'}^{d}_{\mathrm{D}\prec d}$:{NP}

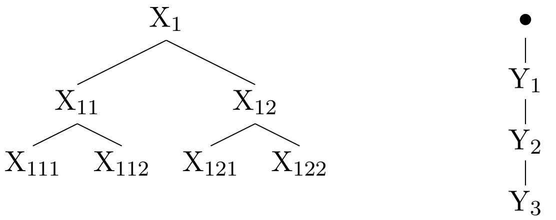
e. $\mathrm{AdvP}^{m}_{m\prec \mathrm{A}}$:{NP, PP, VP}

For C_1 to attach as a left daughter within C'_2, there must be congruence between the sets $\mathrm{DP}^{m}_{D\prec m}$ (93a) and $\mathrm{AdvP}^{d}_{d\prec \mathrm{ADV}}$ (93b). For C'_2 to attach as a right daughter within C_1, there must be congruence between either of the sets $\mathrm{DP}^{d}_{D\prec d}$ (93c) or $\mathrm{D'}^{d}_{D\prec d}$ (93d) and $\mathrm{AdvP}^{m}_{m\prec \mathrm{ADV}}$ (93e). Neither of these constraints is met, and yet the utterance is clearly grammatical.

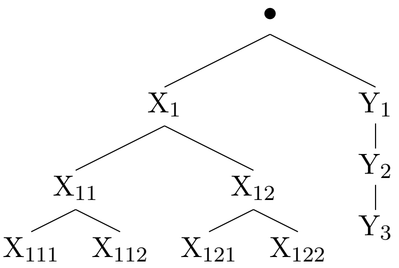
An initial proposal to resolve this is to allow for the introduction of an underspecified node as the mother of C'_{n+1}, as the minimal additional structure necessary for attachment. This node is a variable representing one or more nodes that will be instantiated by the projection of a head word from later in the utterance: it attaches along the right edge of C_n, using one of the modes of attachment described in (72)[22].

Example outcomes of this attachment are shown in (94)

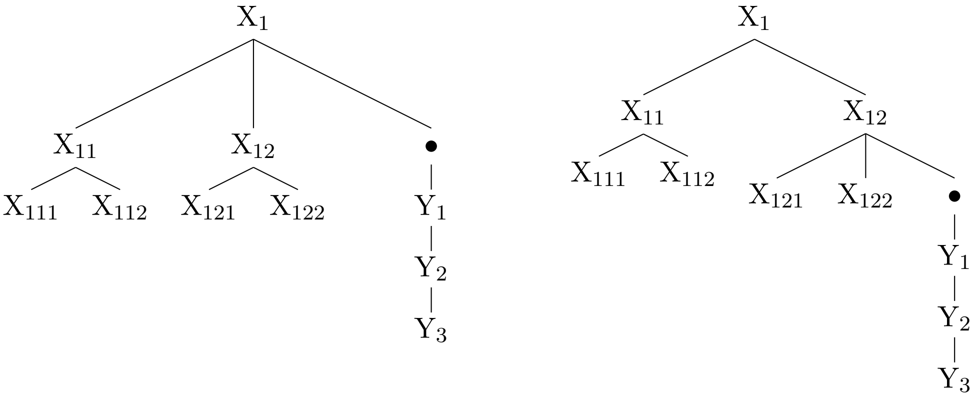
(94)



a. Attachment *above* the right edge: the root node X_1 of C_n and the root node Y_1 of C'_{n+1} are both daughters of the underspecified node.



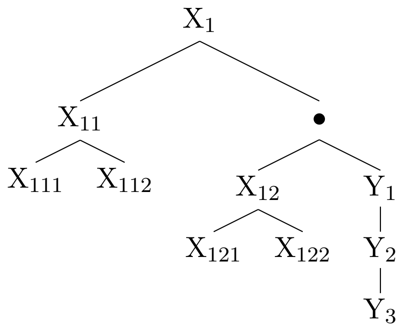
b. Attachment *at* the right edge: the underspecified node above C'_{n+1} attaches as the rightmost daughter of a node n which is member of R_n, with all other daughters of n remaining in the set $M^{-1}(n)$.

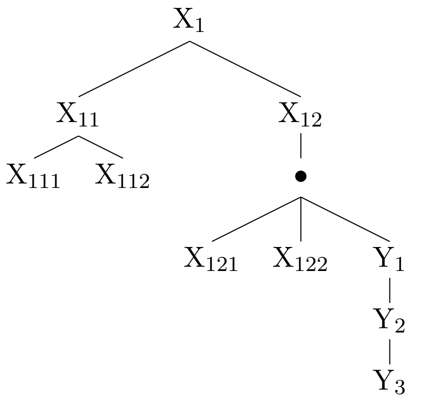


c. Attachment *within* the right edge: the underspecified node C'_{n+1} attaches as the rightmost daughter of a node n which is member of R_n, with at least one other daughters of n becoming a daughter of a node within C'_{n+1}.

---

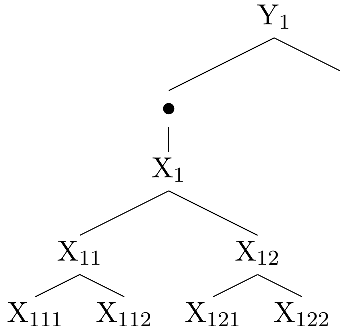
[22] Parallels with this proposal can be seen in the unfixed node used in Dynamic Syntax analyses, whose relation to the rest of the tree is underspecified (Cann et al., 2005, p. 59).





Note that it is not possible to introduce an underspecified node as a left daughter of C'_{n+1} as shown in (95). The underspecified node is not part of the right edge set R_{n+1}, and so cannot be instantiated by a subsequently-added node. To do this would require a terminal node to be added to the left of Y_3 and thus contravene the constraint that the precedence order of the terminal nodes of c-structure must match the order of the s-string.

(95)

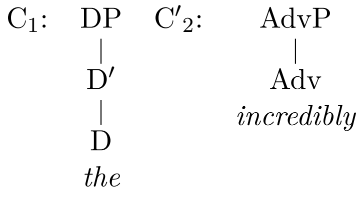


Instantiation of the underspecified node is constrained by the mother and daughter sets at its attachment site. In addition to being a legitimate succeeding mother of Y_1, it must also be a legitimate succeeding daughter and/or mother for the attaching node(s) in X. Thus in (94c), it must have X_1 as a possible left daugher. In (94b) X_1 or X_{12} must be possible preceding mothers. And in (94c), as well as the constraints on the mother, node X_{12}, or nodes X_{121} and X_{122} must be allowable as preceding daughters. These constraints are only tested once the node is instantiated: at the point of attachment, the only c-structure constraint is the requirement not to interrupt an X-bar projection.

Testing this proposal with natural language data shows that adding an underspecified node can allow an incremental analysis of some grammatical strings that would otherwise not be parsable, but that there is no guarantee that a particular choice of where to introduce the node will result in successful structure growth. Furthermore, there are some cases where instantiation of the underspecified node requires the assumptions of additional specified nodes beyond those that are lexically introduced.

Returning to the string *the incredibly inept politician*, adding the second word *incredibly* gives us the situation in (96).

(96)



The relevant mother and daughter sets in (97) show that attachment of C_1 as the left daughter of C'_2 is not possible as the intersection of the two sets is the empty set.

(97) a. $\text{DP}^{m}_{\text{D}\prec m}$: {CP, DP, IP, S}

b. $\text{AdvP}^{d}_{d\prec \text{ADV}}$: { AdvP }

Similarly, attachment of C'_2 as the right daughter of C_1 is not possible, as shown by the relevant mother and daughter sets in (98). Again, the intersection of (98a) with either (98b) or (98c) is the empty set.

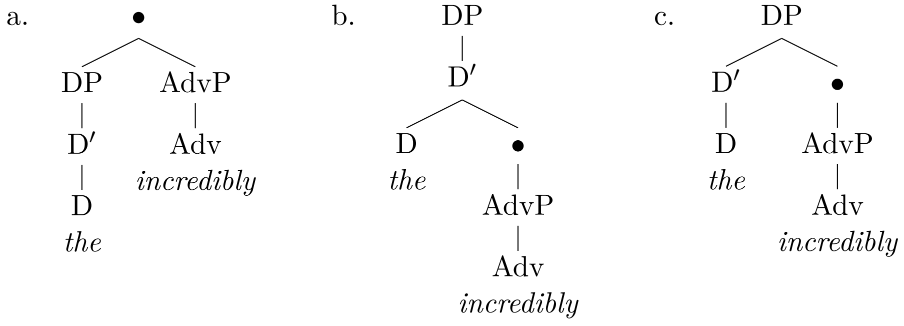
(98) a. $\text{AdvP}^{m}_{m\prec \text{ADV}}$: { NP, PP, VP }

b. $\text{DP}^{d}_{\text{D}\prec d}$: $\emptyset$

c. $\text{D}'^{d}_{\text{D}\prec d}$: { NP }

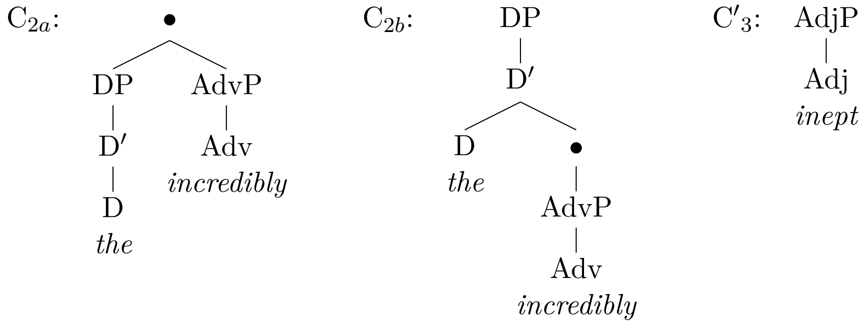
Accordingly, an underspecified node is required for attachment, with the three possibilities for its placement shown in (99).

(99)



Possibility (99c) can be ruled out because the post-head daughter set of DP (98b) is the empty set and so an instantiation here can never be possible. However, there is no a priori reason to choose between possibilities (99a) and (99b). Accordingly we need to pursue both options when adding the next word *inept* to the structure, giving the situation in (100).

(100)



Consider first the option C_{2a}. C'_3 cannot attach above the underspecified node, as this prevents the node from being instantiated. Accordingly, the AdjP maximal projection of C'_3 could instantiate the node, could be the right daughter of the AdvP node in C_{2a}, or could attach within the right edge between the underspecified node and the AdvP node.

If the AdjP is to instantiate the node, its pre-head daughter set must include both Adv and D, and the pre-head mother sets of DP and AdvP must both include Adj. As can be seen from (101), this is not the case: although AdvP could attach as the pre-head daughter of AdjP, this is not possible for DP.

(101) a. $\mathrm{AdjP}^{d}_{d\prec\mathrm{ADJ}}$: { AdvP }

b. $\mathrm{DP}^{m}_{\mathrm{D}\prec m}$: { CP, DP, IP, S }

c. $\mathrm{AdvP}^{m}_{\mathrm{ADV}\prec m}$: { AdjP, AdvP, CP, IP, PP, VP }

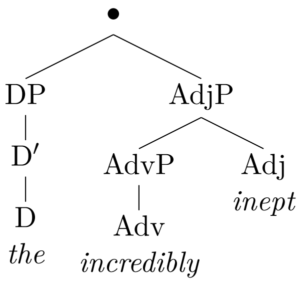
There is similarly no congruence between the post-head daughter set of AdvP and the post-head mother set of AdjP (102), and so C_3 cannot attach as the right daughter of the AdvP in C_{2a}.

(102) a. $\mathrm{AdvP}^{d}_{\mathrm{ADV}\prec d}$: { CP, IP, PP, S }

b. $\mathrm{AdjP}^{m}_{m\prec\mathrm{ADJ}}$: { NP, PP, VP }

However, because the AdvP may be a pre-head daughter of AdjP, C'_3 may be inserted into the right edge of C_{2a}, resulting in the structure in (103).

(103)



Turning now to the possibility of attaching C_3 to C_{2b}, again the attachment may not take place above the underspecified node. Accordingly, C'_3 either attaches as a right daughter of AdvP, or as the right daughter of the underspecified node, or its maximal projection AdjP instantiates the underspecified node. As was shown in (102), AdjP cannot attach as the right daughter of AdvP. However, attachment as the right daughter of the underspecified node is possible: it places a further constraint on the instantiation of the node, but as the intersection of the pre-head mother sets of AdvP and AdjP is not the empty set (104), this possibility cannot be excluded.

(104) a. $\mathrm{AdvP}^{m}_{\mathrm{ADV}\prec m}$: { AdjP, AdvP, CP, IP, PP, S, VP }

b. $\mathrm{AdjP}^{m}_{\mathrm{ADJ}\prec m}$: { CP, IP, NP, S }

c. $\mathrm{AdvP}^{m}_{\mathrm{ADV}\prec m} \cap \mathrm{AdjP}^{m}_{\mathrm{ADJ}\prec m} = \{\ \mathrm{CP, IP, S}\ \}$

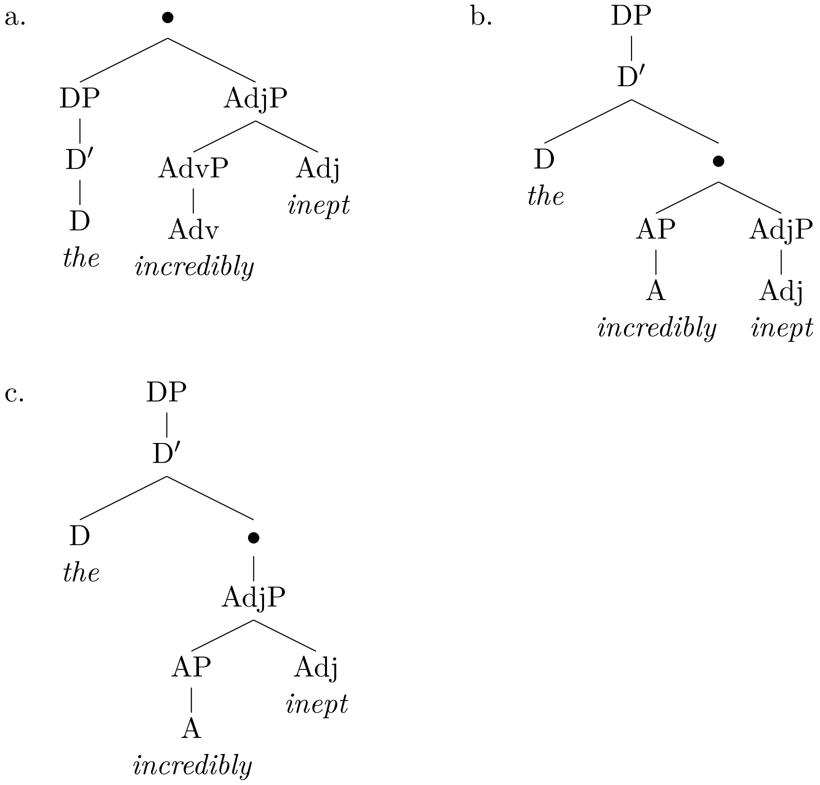
From (104a) we see that AdjP may be a post-head mother of AdvP and so to that extent, AdjP can instantiate the underspecified node in C_{2b}. However, there is no congruence between the pre-head mother set of AdjP and the post-head daughter set of D′ (105). Thus if AdjP is to instantiate the underspecified node in C_{2b}, it requires an underspecified node to be introduced between itself and the D′ node.

(105) a. $\text{AdjP}^{m}_{m\prec\text{ADJ}}$: { VP }

b. $\text{D}'^{d}_{\text{D}\prec d}$: { NP }

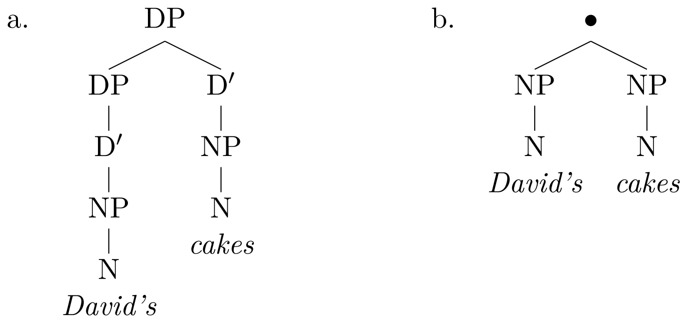
Thus from the situation in (100) we have three possible options for c-structure C_3, shown in (106). We may intuitively be drawn to a preference for (106c) but there is nothing in the internal constraints on c-structure growth that supports that preference.

(106)



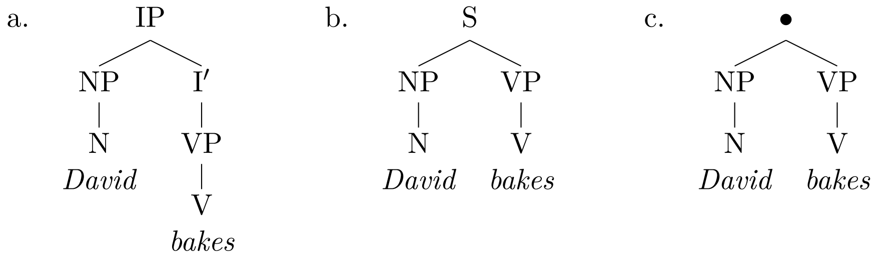
There are also situations where it may not be possible to derive a grammatical c-structure by instantiating an underspecified node, because the intended structure includes either a headless functional phrase or the exocentric category S, both of which are within the grammar of English proposed by Falk (2001). Consider the phrase *David's cakes.* This is analysed in Falk's grammar as having the c-structure in (107a), but because there is no way in the phrase structure rules for an NP to attach directly to an NP, the incrementally-derived structure is that in (107b). Because new structure is introduced along the right edge of (107b), there is no way in which the desired intermediate nodes can be introduced, even if the underspecified node is instantiated with a DP maximal projection.

(107)



Similarly, the sentence *David bakes* has two possible top-down analyses (108a–b) and the incrementally-derived structure (108c).

(108)



Instantiating the underspecified node in (108c) by an IP projected from a word of category I is unsatisfactory because the I will be to the right of the VP, and so both NP and VP are left daughters of IP. There is also no lexical category that can instantiate exocentric S. Thus in this case, it is not possible to generate the desired c-structure incrementally from the lexical input and the parsimonious constraints on c-structural growth that have been assumed.

This could be seen as a reason to relax the constraints on c-structure growth, but this would increase the number of options available at any given point, and thus reduce the predictive power of the theory. Instead, I propose to consider the role of f-structure constraints during incremental growth. These act to constrain the attachment options for unspecified nodes, and also allow for multiple possible c-structures to be abstracted into a single f-structure representation.

3.3.2 Incremental growth of f-structure

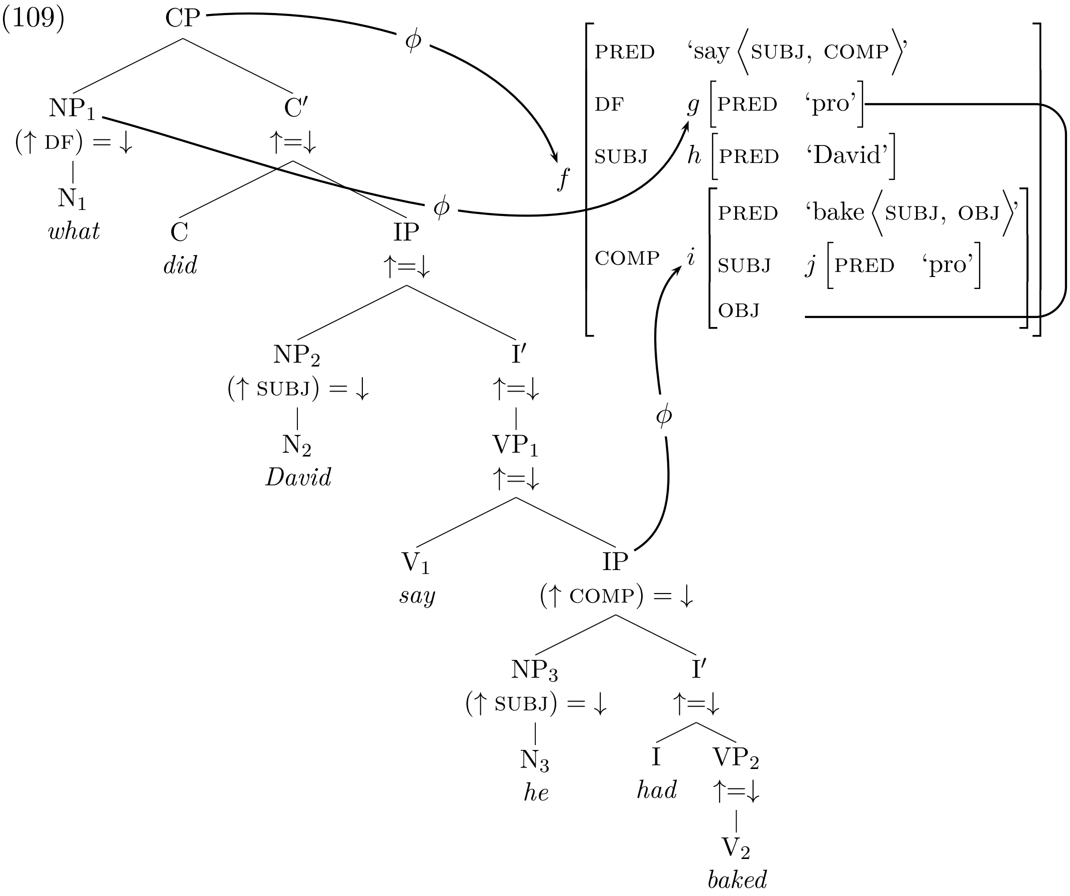
The previous discussion included only c-structure, but an LFG f-structure is always projected from c-structure and forms a part of the syntactic analysis. As a reminder, in the discussion that follows I am considering languages where the c-structure annotation ↑=↓ applies only to co-heads, and where any lexical assignment of grammatical function is only local, e.g. (SUBJ ↑), and without case stacking[23].

---

[23] Case stacking is defined by Nordlinger (1998, p. 5) as occurring "where a single nominal carries multiple case markers, each one indicating a higher relationship within the clause." A fuller explanation showing how constituents with stacked case project f-structure can be found in Frank (2003). The consequences of case-stacking for this theory are left for future work.

The implication of this is that the dominance relationship between c-structure nodes is reflected in the f-structure containment relationship for lexically inserted information. This parallel between c-structure dominance and f-structure containment holds unless an f-structure is subject to structure sharing. In example (109), f-structure g is both f DF and i OBJ. F-structure f is projected by the CP node, and this node dominates NP_1, which projects f-structure g. However, NP_1 is not dominated by node IP, which projects f-structure i.

(109)

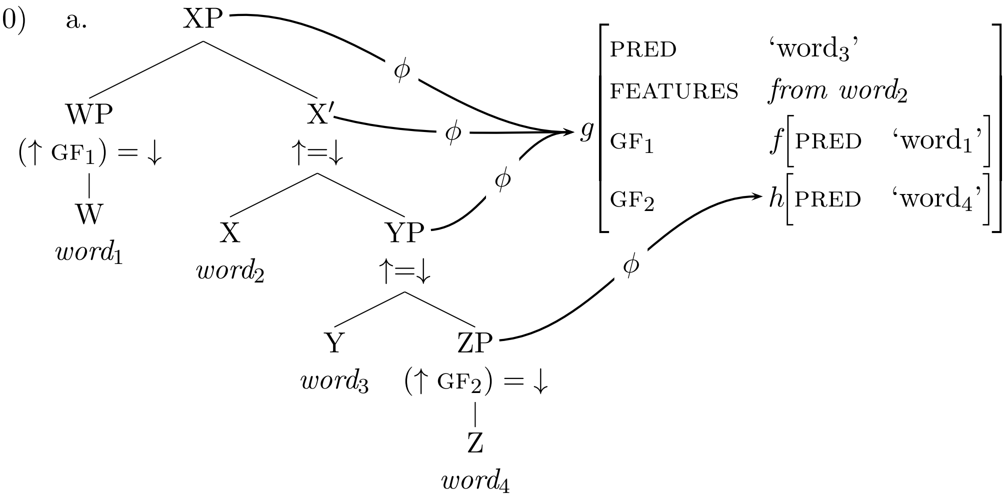


C-structure growth involves one or two nodes within R, the set of nodes on the right edge of the c-structure. Because of the parallels between c-structure dominance and f-structure containment, for an f-structure f, if no members of $\phi^{-1}(f)$ are also members of R, f is closed and it is not possible to add lexical information directly to that f-structure. The only way in which additional grammatical functions can be added to a closed f-structure is via a long-distance dependency.

F-structure growth occurs in two ways. One is the projection of f-structure from c-structure growth as words are attached. The other is induction of f-structure from lexically introduced functional constraints. I will consider growth in projected f-structure first.

The f-structures within the right edge set of nodes R represent the path from the outermost f-structure to the f-structure to which the last word added lexical information. Thus for the c-/f-structure pair in (110a), the path representing R_4 is given in (110b). The ϕ-function mapping is shown only for nodes in R.

(110) a.



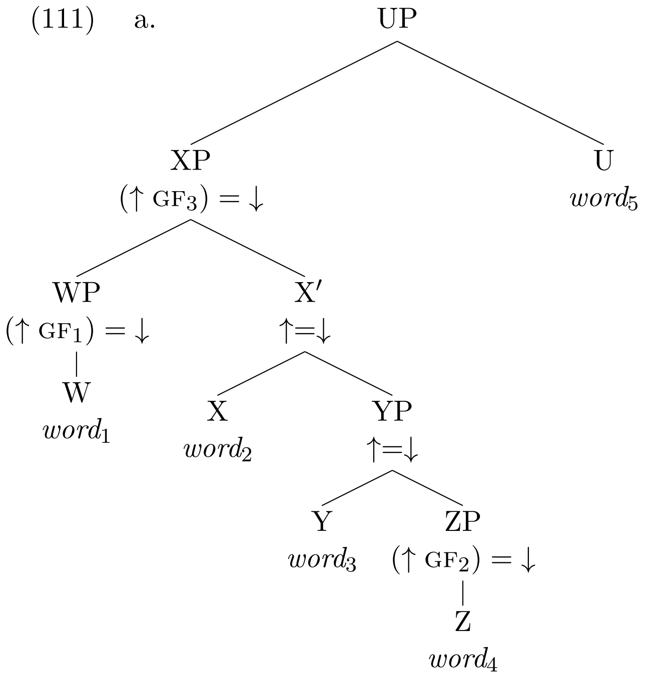
b. F-structures projected from $R_4 = g\ h$

In Section 3.3.1, I showed that constraints on c-structure result in six possibilities for incremental growth: three types of attachment each either with or without an underspecified node. I now show how each of these possibilities is reflected in the growth of f-structure. This is done in relation to example (110a). In each case the c- and f-structure pair resulting from attachment is shown, together with the set of f-structures projected by the right edge set of nodes R_4 before the attachment of *word*$_4$ and by the right edge set of nodes R_5 after *word* $_5$ is attached.

3.3.2.1 Direct attachment above

If C_{n+1} attaches above C_n, becoming its mother, the lexical information from C_{n+1} is added either to a new f-structure that contains the previous outermost f-structure, or to the outermost f-structure if C_n and C_{n+1} are co-heads. The case where a new f-structure is added that contains the previous outermost f-structure is shown in (111a).

(111) a.



$$i\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_5\text{’} \\ \textsc{gf}_3 & g\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_3\text{’} \\ \textsc{features} & \textit{from word}_2 \\ \textsc{gf}_1 & f\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_1\text{’} \end{array}\right] \\ \textsc{gf}_2 & h\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_4\text{’} \end{array}\right] \end{array}\right] \end{array}\right]$$

b.

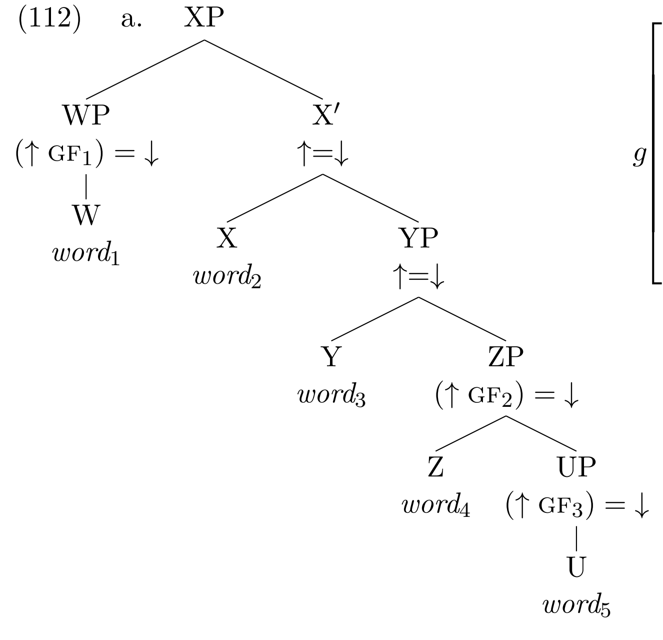
	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = i$

After attachment, f-structure h is closed.

3.3.2.2 Direct attachment at the right edge

If the attachment point is within the phrasal chain from *word*$_n$ which projects f-structure h, then *word*$_{n+1}$ adds information to f-structure h or to an f-structure contained by it (112a). Otherwise, information is added to an f-structure that f-commands h (113a-114a). Note that in these examples, the f-structures in (113a) and (114a) are identical, despite the different c-structures. However, the value of the variable GF$_3$ may instantiate differently between the examples.

(112) a.

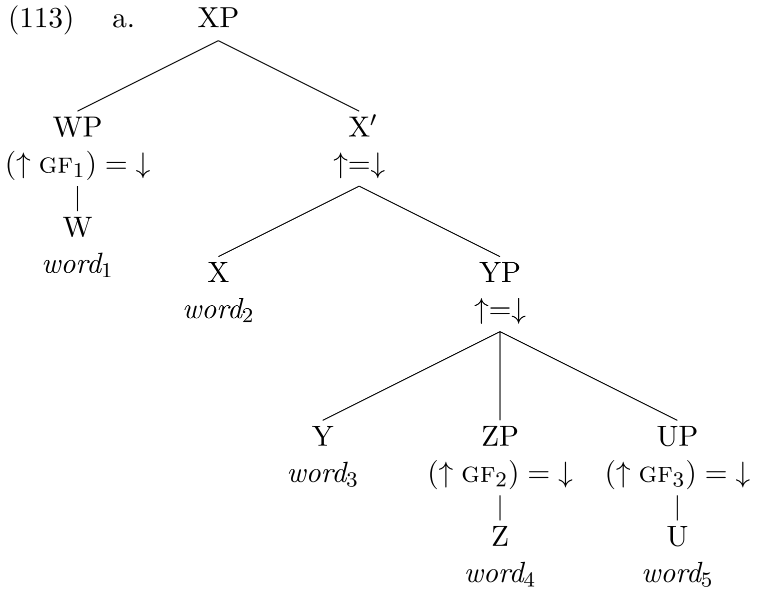


$$g\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_3\text{’} \\ \textsc{features} & \textit{from word}_2 \\ \textsc{gf}_1 & f\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_1\text{’} \end{array}\right] \\ \textsc{gf}_2 & h\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_4\text{’} \\ \textsc{gf}_3 & i\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_5\text{’} \end{array}\right] \end{array}\right] \end{array}\right]$$

b.

	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = g\ h\ i$

(113) a.

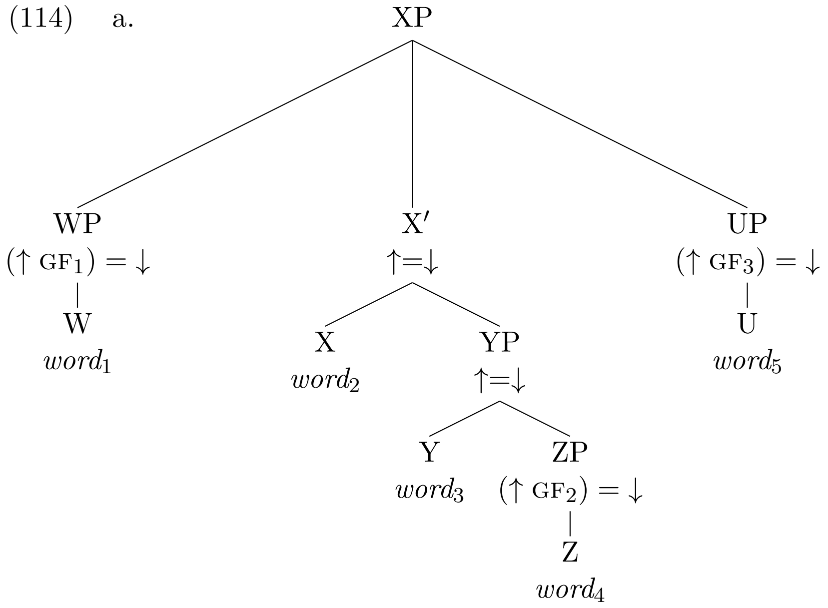


$$g\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_3\text{’} \\ \textsc{features} & \textit{from word}_2 \\ \textsc{gf}_1 & f\left[\textsc{pred}\quad \text{‘word}_1\text{’}\right] \\ \textsc{gf}_2 & h\left[\textsc{pred}\quad \text{‘word}_4\text{’}\right] \\ \textsc{gf}_3 & i\left[\textsc{pred}\quad \text{‘word}_5\text{’}\right] \end{array}\right]$$

b.

	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = g\ i$

(114) a.



$$g\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_3\text{’} \\ \textsc{features} & \textit{from word}_2 \\ \textsc{gf}_1 & f\left[\textsc{pred}\quad \text{‘word}_1\text{’}\right] \\ \textsc{gf}_2 & h\left[\textsc{pred}\quad \text{‘word}_4\text{’}\right] \\ \textsc{gf}_3 & i\left[\textsc{pred}\quad \text{‘word}_5\text{’}\right] \end{array}\right]$$

b.

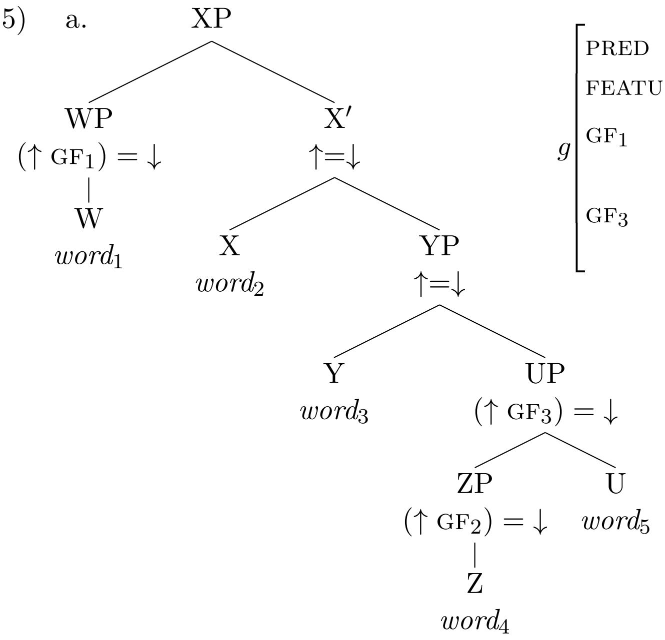
	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = g\ i$

In example (112a), f-structure h remains open after attachment of *word*$_5$, but in (113a) and (114a), f-structure h is closed by the attachment.

3.3.2.3 Direct attachment within the right edge

The new information is added to a new f-structure somewhere between the outermost f-structure and the f-structure that immediately contains h (115a-116a).

(115) a.

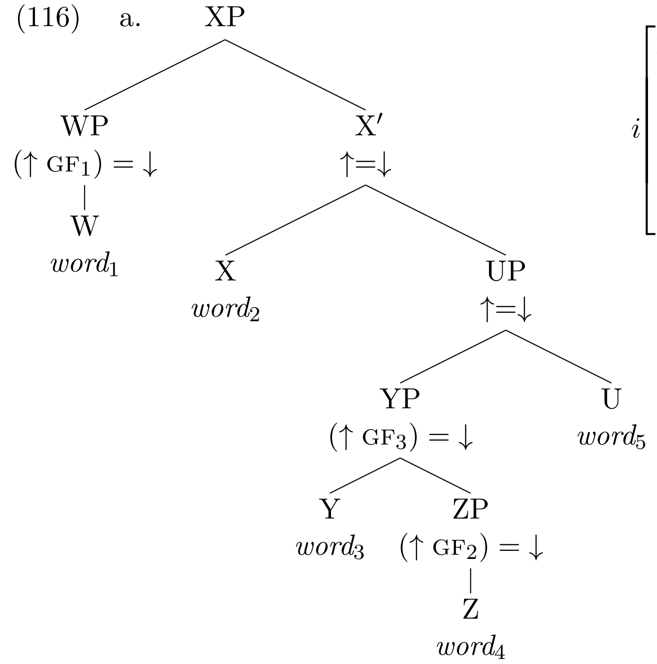


$$g\begin{bmatrix} \textsc{pred} & \text{`word}_3\text{'} \\ \textsc{features} & \textit{from word}_2 \\ \textsc{gf}_1 & f\begin{bmatrix} \textsc{pred} & \text{`word}_1\text{'} \end{bmatrix} \\ \textsc{gf}_3 & i\begin{bmatrix} \textsc{pred} & \text{`word}_5\text{'} \\ \textsc{gf}_2 & h\begin{bmatrix} \textsc{pred} & \text{`word}_4\text{'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b.

	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = g\ i$

(116) a.



$$i\begin{bmatrix} \textsc{pred} & \text{`word}_5\text{'} \\ \textsc{gf}_1 & f\begin{bmatrix} \textsc{pred} & \text{`word}_1\text{'} \end{bmatrix} \\ \textsc{gf}_3 & g\begin{bmatrix} \textsc{pred} & \text{`word}_3\text{'} \\ \textsc{gf}_2 & h\begin{bmatrix} \textsc{pred} & \text{`word}_4\text{'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b.

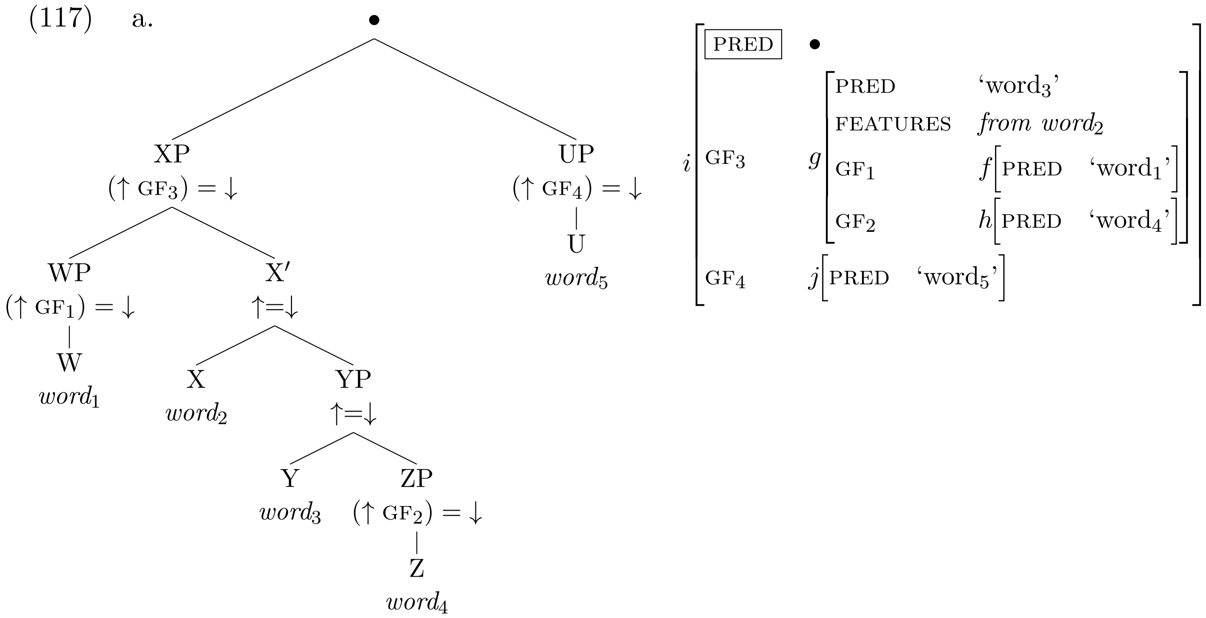
	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = i$

In both of these cases, f-structure h is closed by the attachment.

3.3.2.4 Attachment to an underspecified node above

If C_n and C_{n+1} are both daughters of an underspecified node, then the new information is added to a structure which — at that point — f-commands all of the f-structures projected from C_n. A further new f-structure is projected that contains the C_n and C_{n+1} f-structures, as shown in example (117a).

(117) a.



b.

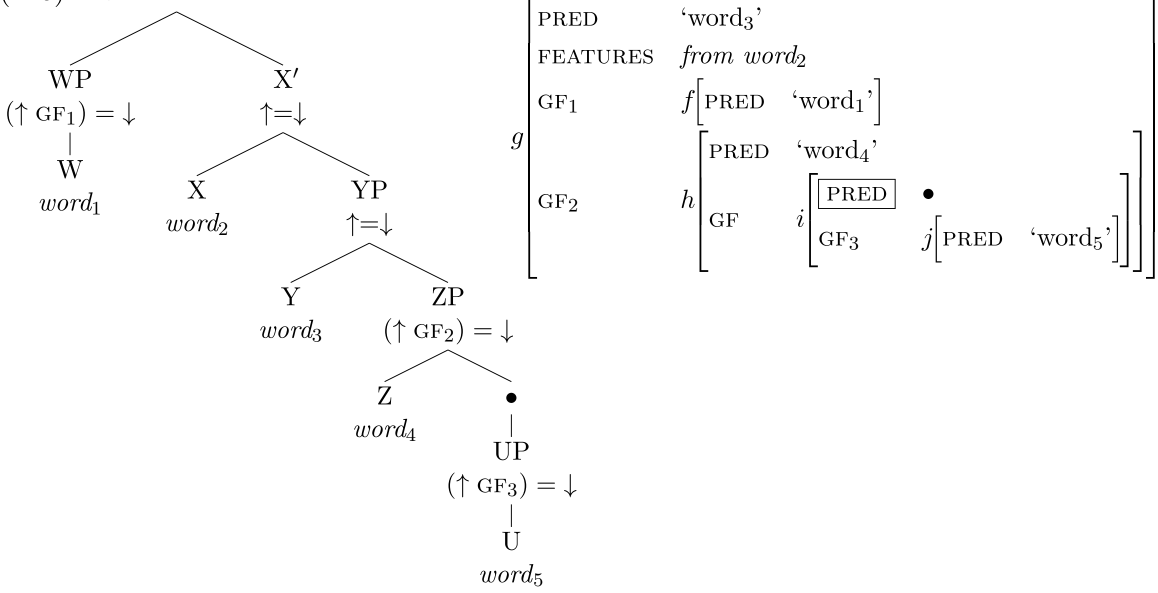
	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = i\ j$

F-structure h is closed by the attachment.

3.3.2.5 Attachment to an underspecified node at the right edge

New information is added to an f-structure that is a value of an otherwise empty f-structure that is either contained by the previously added f-structure, as in (118a) or which f-commands it, as in (119a).

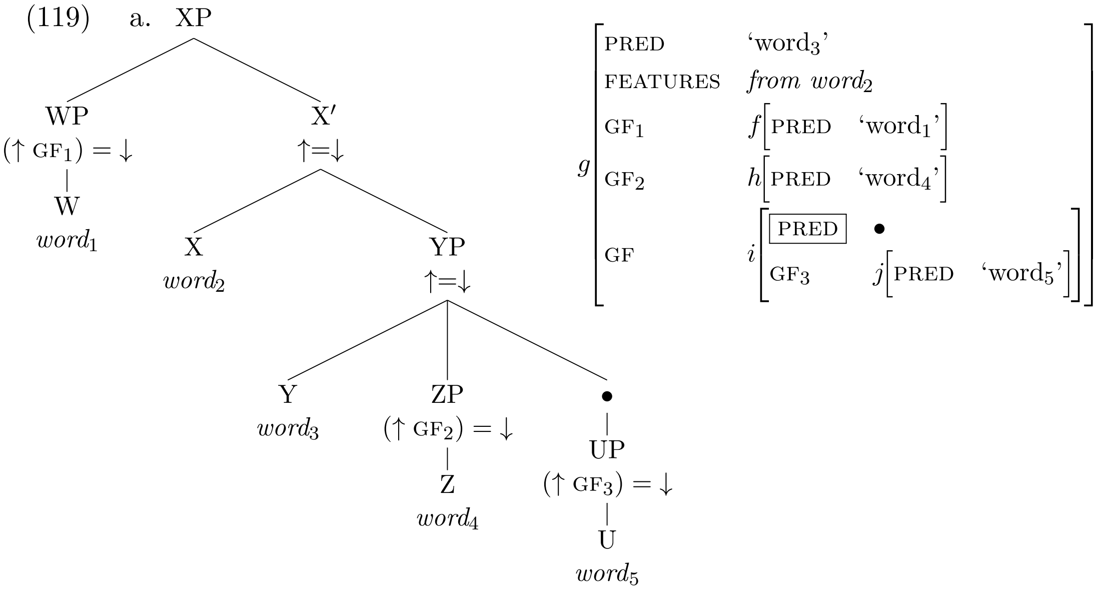
(118) a.



b.

	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = g\ h\ i\ j$

(119) a.



b.

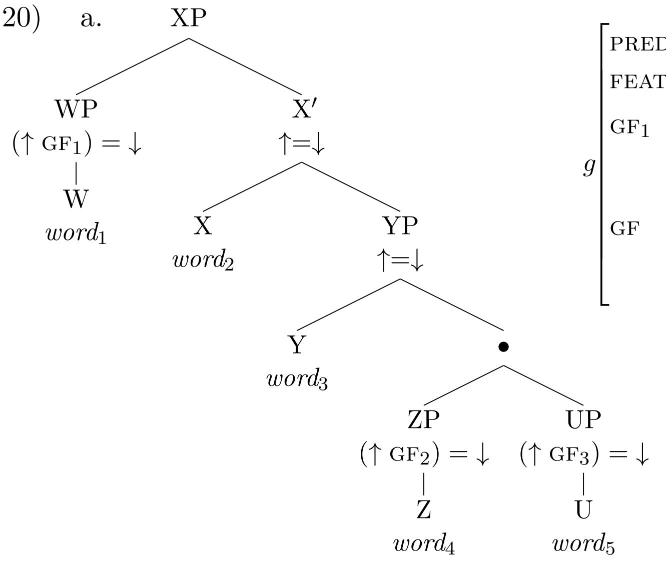
	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = g\ i\ j$

In both cases, again, f-structure h is closed by the attachment.

3.3.2.6 Attachment to an underspecified node within the right edge

The f-structure projected by the underspecified node contains at least f-structure h and possibly other f-structures that contain h (120a).

(120) a.



$$g\left[\begin{array}{ll} \textsc{pred} & \text{‘word}_3\text{’} \\ \textsc{features} & \textit{from word}_2 \\ \textsc{gf}_1 & f\left[\textsc{pred}\quad \text{‘word}_1\text{’}\right] \\ \textsc{gf} & i\left[\begin{array}{ll} \boxed{\textsc{pred}} & \bullet \\ \textsc{gf}_2 & h\left[\textsc{pred}\quad \text{‘word}_4\text{’}\right] \\ \textsc{gf}_3 & j\left[\textsc{pred}\quad \text{‘word}_5\text{’}\right] \end{array}\right] \end{array}\right]$$

b.

	After *word*$_4$	After *word*$_5$
F-structures projected	from $R_4 = g\ h$	from $R_5 = g\ i\ j$

Once again, f-structure h is closed by the attachment.

3.3.2.7 Growth in f-structure by induction from lexical constraints

Lexical specifications can introduce constraints on functional structure. In order for these constraints to be recognised at the point that the word is added to the utterance, it may be necessary for f-structures to be included that are not directly projected from a c-structure node, but that are induced from lexical content.

Consider the incremental growth of the two Korean sentences in (121).

(121) a. *pi-ka* *ecey* *o-asseyo*
rain-SUBJ yesterday come-PST.POL
"It rained yesterday."

b. *ecey* *pi-ka* *o-asseyo*
yesterday rain-SUBJ come-PST.POL
"Yesterday it rained."

Example (52) from Section 3.1.5.1, repeated here, shows the lexical specification for Korean *pika* 'rain.SUBJ'.

(52) *pika* N (↑ PRED) = 'rain'
(SUBJ ↑)

The f-structure projected from *pika* provides the SUBJ grammatical function within another f-structure. By induction from the principle of coherence, this second f-structure must also contain a PRED value. Accordingly, at the point that *pika* is added to c-structure, an f-structure with an empty PRED value must be introduced if one is not already present, as in example (122). The analysis follows the phrase-structure rules proposed by Cho and Sells (1995), according to which the maximal projection of lexical heads is X′[24].

---

[24] The authors acknowledge that this is an unusual assumption and do not defend it any further.

(122)

$$\begin{bmatrix} \boxed{\textsc{pred}} & \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{`rain'}\end{bmatrix}\end{bmatrix}$$

F-structures introduced in this way are equivalent to those projected by underspecified c-structure nodes. If an underspecified c-structure node is subsequently required immediately above the phrase from which the PRED-less f-structure was induced, it projects the outer f-structure. Compare (124) and (125), which show *pika* 'rain.SUBJ' being added to *ecey* 'yesterday' and vice versa. The lexical specification of *ecey* in (123) does not introduce a functional constraint, and so the outer f-structure in (124b) is projected from the underspecified node required for attachment. However, the outer f-structure in (125b) was already present at (125a) and no additional f-structure is added by the underspecified node.

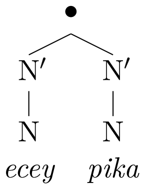
(123) *ecey* N (↑ PRED) = 'yesterday'

(124) *ecey* *pi-ka* ...
yesterday rain-SUBJ ...
(partial) "Yesterday it rain(ed)"

a.

$$\begin{bmatrix}\textsc{pred} & \text{`yesterday'}\end{bmatrix}$$

b.



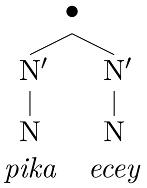
$$\begin{bmatrix} \boxed{\textsc{pred}} & \bullet \\ \textsc{gf} & \begin{bmatrix}\textsc{pred} & \text{`yesterday'}\end{bmatrix} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{`rain'}\end{bmatrix}\end{bmatrix}$$

(125) *pi-ka* *ecey* ...
rain-SUBJ yesterday ...
(partial) "It rain(ed) yesterday"

a.

$$\begin{bmatrix} \boxed{\textsc{pred}} & \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{`rain'}\end{bmatrix}\end{bmatrix}$$

b.

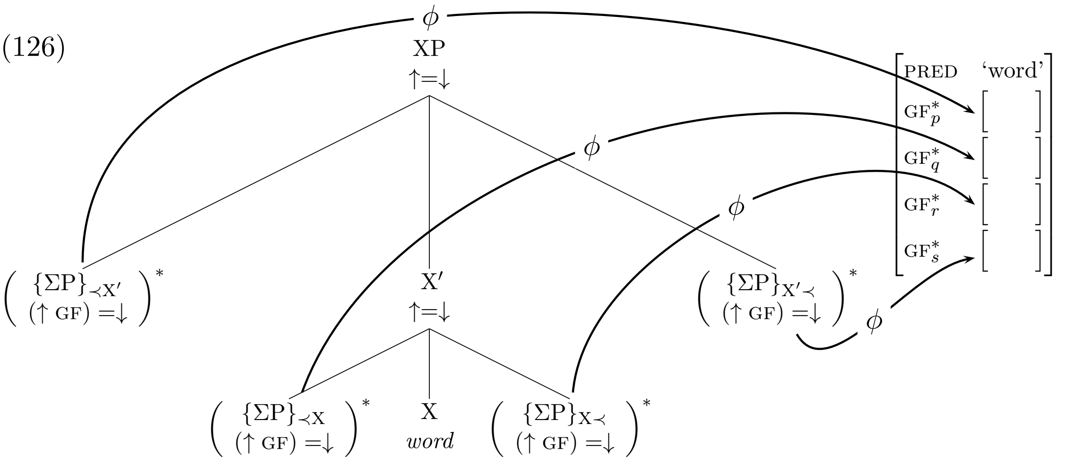


$$\begin{bmatrix} \boxed{\textsc{pred}} & \bullet \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{`rain'}\end{bmatrix} \\ \textsc{gf} & \begin{bmatrix}\textsc{pred} & \text{`yesterday'}\end{bmatrix}\end{bmatrix}$$

3.3.2.8 Reflecting phrase-structure constraints in f-structure

Phrase-structure rules may impose constraints on the ordering of dependents relative to each other and the phrasal head. Where present, these are reflected in f-structure by the order in which dependents are added. Considering a general phrase-structure rule in (73) which includes both the X′ and XP projections of lexical category X, this generates the potential c- and f-structure pair for the maximal phrase in (126).

(126)



Elements of an f-structure are not ordered. However, when f-structure is built incrementally, the phrase-structure rules impose constraints on the order in which the values of attributes are added to an f-structure. This is indicated in (126) by the labels p, q, r, s on the grammatical functions in f-structure. In other words, for each category X we can derive a grammatical function ordering as in (127).

(127) X: $\text{GF}_p^* \prec \text{GF}_q^* \prec \text{PRED} \prec \text{GF}_r^* \prec \text{GF}_s^*$

The exact nature of the ordering depends on the language-specific rules, and ordering constraints in phrase-structure rules are by no means universal. There may be complete freedom of the ordering of dominated phrases, or there may be partial constraints. The ordering may be partial, such that several members of the set may appear at a particular position, or one member of the set may appear at several positions. A grammatical function may also be licensed to appear in more than one position in a phrase-structure rule. This might be where the grammatical function is represented by a set such as the non-governed ADJ grammatical function, or with discontinuous constituents in a language with free word order. Because of this a number of potential orderings can be derived from the higher-level ordering.

Tables 3.2 and 3.3 showed the possible mother-daughter combinations from the English phrase structure rules in Appendix A, but the only element of ordering derivable from the tables was whether the daughter attached before or after the head of the mother phrase. Tables 3.4 and 3.5 below present richer information, including the grammatical function(s) that a daughter category may fulfil in relation to its mother. In the tables the symbol ↑=↓ represents a daughter category that is a co-head with its mother.

Table 3.4: Daughter phrases and GFs permitted before mother head

Daughter phrases: Succeeding mother sets $\mathrm{X}^m_{\mathrm{X}\prec m}$: read along rows	Mother node categories Preceding daughter sets $\mathrm{X}^d_{d\prec \mathrm{X}}$: read down columns									
	CP	C′	IP	I′	DP	D′	AP	NP	PP	VP
CP			SUBJ							
IP										
S										
DP	DF*		DF† SUBJ		POSS‡					
AdjP	DF*		DF†					ADJ		
AdvP	DF*		DF†				ADJ		ADJ	ADJ
NP	DF*		DF† SUBJ							
PP	DF*		DF† SUBJ							ADJ
VP	DF*		DF†							

Ordering constraint: DF$\prec$SUBJ
* Must contain a "wh-pronoun"
† Must not contain a "wh-pronoun"
‡ Daughter DP must carry GEN case.

Applying the ordering constraints allows us to derive partially ordered daughter sets that include information on the grammatical function that can be provided by a particular daughter category: members of the set are pairs whose first member is a grammatical function and whose second member is a set of categories. For example, reading down the IP column of Table 3.4 gives us the preceding daughter set for IP in (128).

(128) $\mathrm{IP}^d_{d\prec \mathrm{I}}$: [⟨DF, {AdjP, AdvP, DP, NP, PP, VP}⟩,
⟨SUBJ, {AdjP, AdvP, CP, DP, NP, PP, VP}⟩]

Similarly, reading down the VP column of Table 3.3 gives us the succeeding daughter set for VP in (129).

(129) $\mathrm{VP}^d_{\mathrm{V}\prec d}$: [⟨OBJ, {DP, NP}⟩,
⟨OBJ$_\theta$, {DP, NP}⟩,
⟨XCOMP, {AdjP, VP}⟩,
⟨OBL, {PP}⟩,
⟨COMP, {CP, IP, S}⟩,
⟨ADJ, {AdjP, PP}⟩]

The succeeding daughter sets derived from Table 3.3 are the general case for each phrasal category. Because the daughter phrases follow the head of the mother phrase, the selectional restrictions of the mother are known, and so in practice the daughter set is

Table 3.5: Daughter phrases and GFs permitted after mother head

Mother node categories Succeeding daughter sets $X^{d}_{X\prec d}$: read down columns										Daughter phrases: Preceding mother sets $X^{m}_{m\prec X}$: read along rows
CP	C′	IP	I′	DP	D′	AP	NP	PP	VP	
						COMP	COMP ADJ		COMP	CP
	↑=↓					COMP	ADJ		COMP	IP
	↑=↓					COMP	ADJ		COMP	S
								OBJ	OBJ OBJ_θ	DP
									XCOMP	AdjP
								ADJ	ADJ	AdvP
					↑=↓			OBJ	OBJ OBJ_θ	NP
						OBL ADJ	OBL ADJ	OBL ADJ	OBL ADJ	PP
	↑=↓		↑=↓						XCOMP	VP

Ordering constraint: $\text{OBJ}\prec\text{OBJ}_\theta\prec\text{XCOMP}\prec\text{OBL}\prec\text{COMP}\prec\text{ADJ}$

more specific and lexically determined. Where there are valency alternations for a head (e.g. *give* '⟨SUBJ, OBJ, {OBJ_θ | OBL}⟩') these can be reflected in the sets. Examples of more specific succeeding daughter sets are given in (130).

(130) a. *hit* $\text{VP}^{d}_{V\prec d}$: [⟨OBJ, {DP, NP}⟩,
⟨ADJ, {AdjP, PP}⟩]

b. *give* $\text{VP}^{d}_{V\prec d}$: [⟨OBJ, {DP, NP}⟩,
{ ⟨OBJ_θ, {DP, NP}⟩ | ⟨OBL, {PP}⟩ },
⟨ADJ, {AdjP, PP}⟩]

3.3.2.9 F-structure growth and headless phrases

A significant problem with introducing underspecified nodes to c-structure, identified in Section 3.3.1.3, is the impossibility of instantiating the node if the phrase-structure rules require it to be a node within a headless phrase. However adding functional constraints to the categorial attachment constraints can allow the problem to be resolved.

I will explore this using the ordering constraints of English, and consider first the functional categories DP and D′. The only possible attachments to a phrase projected by a word of category D are a GEN case-marked DP as a preceding daughter, attaching at DP, and a co-head NP as a succeeding daughter, attaching at D′. The corollary of this is a GEN case-marked DP is constrained to provide the POSS grammatical function as part of an f-structure where the PRED value is not yet known, but is constrained to be provided by a lexical item of category N. Looking at the categories of which NP can be a daughter (from the NP rows in Tables 3.4 and 3.5), there are three possibilities.

(131) Possibilities for NP attachment as a daughter.

a. An NP can provide the DF or the SUBJ grammatical function to an f-structure projected from a CP or IP node where the PRED value of that f-structure is not yet present.

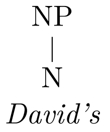
b. An NP can provide the OBJ or OBJ_θ grammatical function to an f-structure projected from a PP or a VP. In these cases, the grammatical function is governed by the PRED, which must already be present in the f-structure.

c. An NP can be a co-head to a DP, providing the PRED value to the f-structure projected by that DP.

Looking at the C′ and I′ columns in Table 3.5, we see that VP is the only lexical category that can function as a co-head and provide a PRED value to the f-structure projected from CP, IP. In other words, when a non-case-marked NP attaches to CP or IP, the PRED value of the f-structure projected by the mother phrase must not yet be present, and that PRED value is constrained to be provided by a lexical head of category V.

Returning to the phrase *David's cakes*, introduced previously at (107), if we project f-structure from the initial NP projected by *David's* and include the attachment constraints, we have the c- and f-structure pair in (132).

(132)

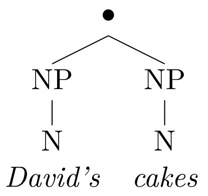


$$g\begin{bmatrix}\text{PRED} & \\ \text{POSS} & f\begin{bmatrix}\text{PRED} & \text{'David'}\\ \text{CASE} & \text{GEN}\end{bmatrix}\end{bmatrix}$$

$$g\ \text{CAT} =_c [\text{ADJ} -,\ \text{ADV} -,\ \text{N} +,\ \text{P} -,\ \text{V} -]$$

Adding the second word of the phrase *cakes* requires an underspecified node above both NPs, and this underspecified node also projects f-structure. If case (131a) applies, we have the situation in (133).

(133)



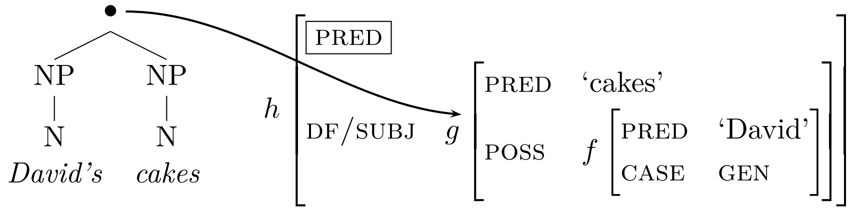
$$g\begin{bmatrix}\boxed{\text{PRED}} & \\ \text{POSS} & f\begin{bmatrix}\text{PRED} & \text{'David'}\\ \text{CASE} & \text{GEN}\end{bmatrix}\\ \text{DF/SUBJ} & h\begin{bmatrix}\text{PRED} & \text{'cakes'}\end{bmatrix}\end{bmatrix}$$

$$g\ \text{CAT} =_c [\text{ADJ} -,\ \text{ADV} -,\ \text{N} +,\ \text{P} -,\ \text{V} -]$$
$$g\ \text{CAT} =_c [\text{ADJ} -,\ \text{ADV} -,\ \text{N} -,\ \text{P} -,\ \text{V} +]$$

There is a feature clash in the attachment constraints arising from the two NPs, and so it is not possible to instantiate the underspecified node.

Case (131b) cannot apply because there is no PRED present to select an OBJ or OBJ$_\theta$ grammatical function, and so we are left with case (131c) where the NP, functioning as co-head to a DP, provides the PRED value to its f-structure. This gives us the situation in (134). The attachment constraint contributed by *cakes* is still present, resulting in the introduction of the PRED-less f-structure h whose PRED value is constrained to be provided by a lexical item of category V.

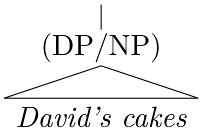
(134)



$$g\ \text{CAT} =_c [\text{ADJ} -,\ \text{ADV} -,\ \text{N} +,\ \text{P} -,\ \text{V} -]$$
$$h\ \text{CAT} =_c [\text{ADJ} -,\ \text{ADV} -,\ \text{N} -,\ \text{P} -,\ \text{V} +]$$

The underspecified node is still not instantiated, but the f-structure it projects is complete, coherent, and consistent. Top-down application of phrase-structure rules allows the node to be identified as a DP, but there is already sufficient information present to constrain incremental attachment. For the purposes of building c-structure, the phrase can be represented as in (135), with a more exact representation being left for later calculation if required.

(135)



3.3.3 Functional constraints on incremental growth

In the discussion above, satisfaction of the principles of completeness, coherence and consistency was used as a criterion for the licensing of c-structures where an underpecified node is not lexically instantiated. In general, functional constraints interact with c-structure well-formedness constraints and phrase-structure rules, to further constrain incremental structural growth.

Temporary incompleteness and incoherence of f-structures is unavoidable during the incremental building of structure. However, ungrammaticality occurs if a point is reached in the structure-building where lexical information can no longer be added to an f-structure, as described in Section 3.3.2. Thus the requirement for any temporary incoherence and incompleteness to be resolved constrains the possible attachment sites within c-structure, and hence the sites in f-structure where growth can occur.

3.3.3.1 The principle of coherence

In Sections 3.3.2.4-3.3.2.7, I showed how underspecified f-structures could be introduced by an underspecified c-structure node or a lexically specified functional constraint, e.g. (SUBJ $\uparrow$).

In the grammars in Appendix A the PRED value for an f-structure is provided directly by lexical information: structure-sharing applies to a whole f-structure which fills more than one grammatical function. Accordingly, if lexical information can no longer be added to a PRED-less f-structure because its projecting nodes are no longer on the right edge of c-structure, the principles of completeness and coherence are irrevocably breached and the utterance is necessarily ungrammatical. This leads to a prediction of a strong constraint on c-structure attachment (136).

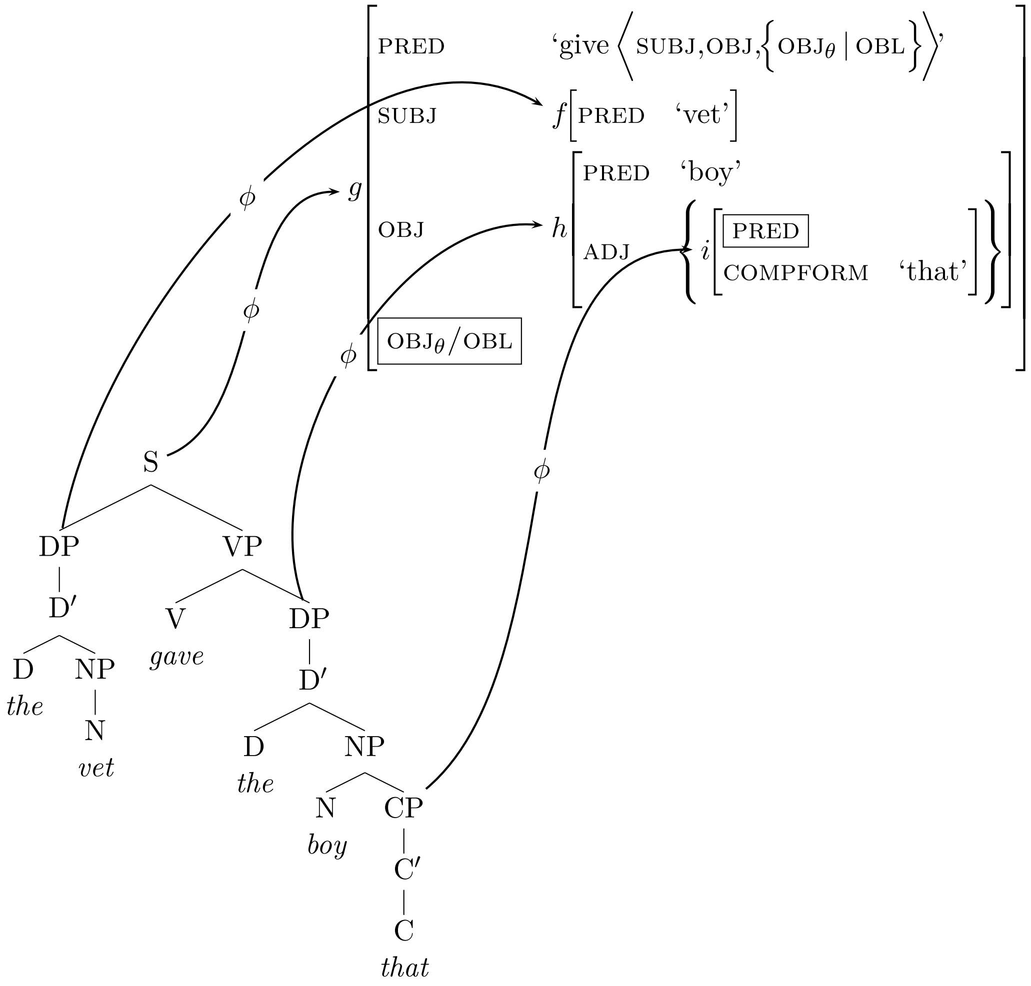
(136) **Prediction 1** If the f-structure projected by c-structure C_n contains an f-structure f without a PRED value, the attachment site must be dominated by the topmost node in $\phi^{-1}(f)$.

An example of this prediction can be seen by considering the sentence in (137).

(137) *The vet gave the boy that the dog bit a gift.*

The c- and f-structure pair generated after the first six words *The vet gave the boy that* is shown in (138). For clarity, only the ϕ-functions from the topmost projecting nodes are shown[25].

(138) *The vet gave the boy that...*

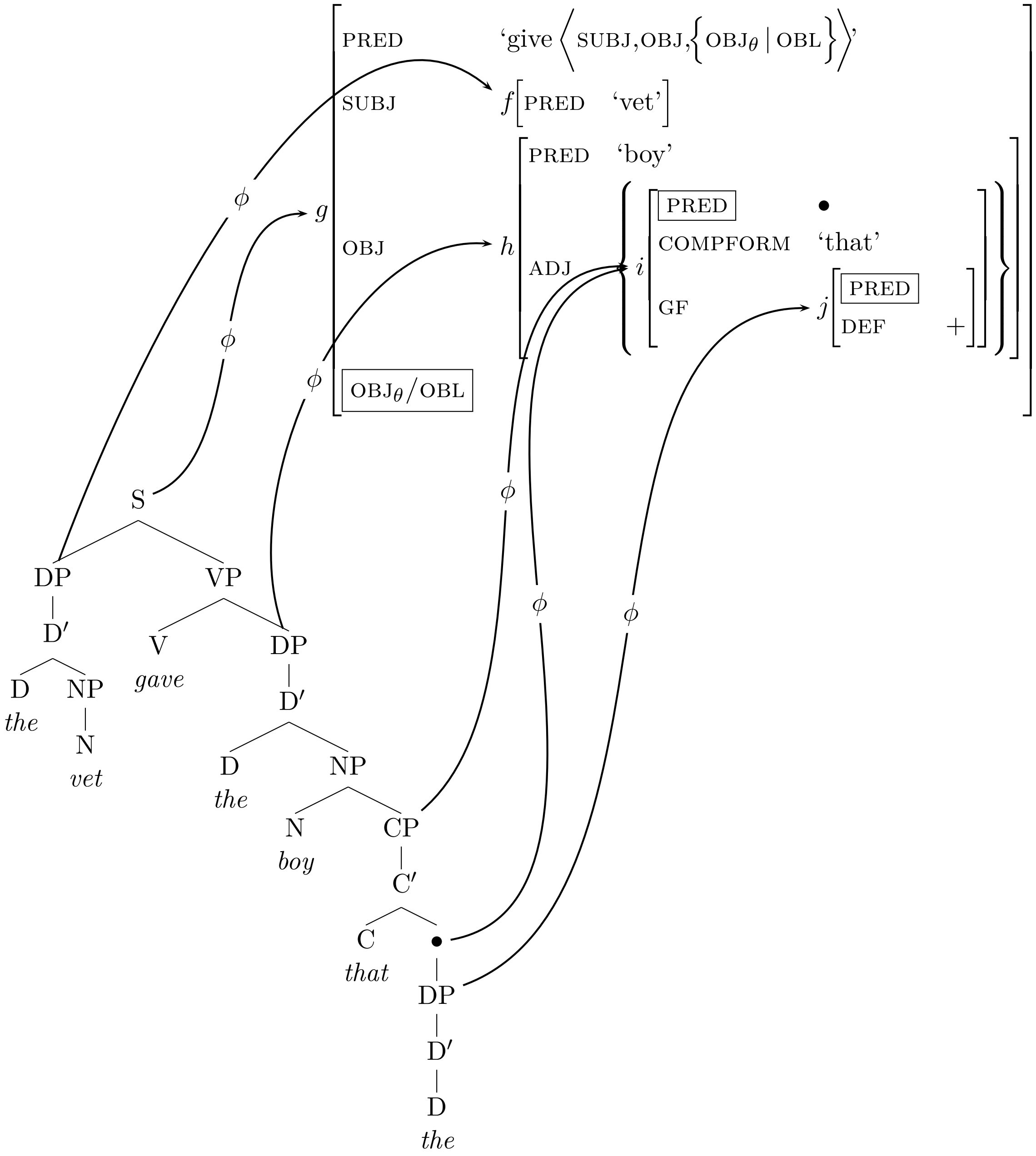


---

[25] I assume that *that* is unambiguously recognised as a complementiser rather than a determiner.

According to the phrase structure rules, the DP projected by the seventh word *the* could attach directly at VP, and indeed there a vacant argument slot in f-structure g for this attachment. However, f-structure i has no PRED value and so following Prediction 1, the DP projected by *the* must attach at a site dominated by the CP node that is the topmost node projecting f-structure i. Accordingly, the DP attaches to an underspecified node that is a daughter of the C′ node, producing the c- and f-structure pair in (139). I return to this example in Section 6.4.1.

(139) *The vet gave the boy that the...*



The principle of coherence can also lead to f-structure growth without the introduction of an unspecified c-structure node. One example of this is the case of Dutch crossing dependencies such as example (140) taken from Bresnan et al. (1982, ex. 26, p. 625).

(140) *...dat Jan Piet Marie zag helpen zwemmen*
...that Jan Piet Marie see.PST help.INF swim.INF
“...that Jan saw Piet help Marie swim”

After the fourth word of the fragment, the f-structure is as shown in (141).

(141)
$$\begin{bmatrix} \boxed{\textsc{pred}} & \\ \textsc{compform} & \text{‘dat’} \\ \textsc{gf} & \begin{bmatrix}\textsc{pred} & \text{‘Jan’}\end{bmatrix} \\ \textsc{gf} & \begin{bmatrix}\textsc{pred} & \text{‘Piet’}\end{bmatrix} \\ \textsc{gf} & \begin{bmatrix}\textsc{pred} & \text{‘Marie’}\end{bmatrix} \end{bmatrix}$$

The fifth word, *zag* “see.PST” has a lexical specification with an optional XCOMP grammatical function[26], shown in (142).

(142)
$$\begin{array}{lll} \textit{zag} & \text{V} & (\uparrow \textsc{pred}) = \text{‘see’} \\ & & (\uparrow \textsc{tense}) = \textsc{pst} \\ & & (\uparrow \textsc{subj}) \\ & & (\uparrow \textsc{obj}) \\ & & \left(\begin{array}{l} (\uparrow\ \textsc{xcomp}) \\ (\uparrow\ \textsc{xcomp subj}) = (\uparrow\ \textsc{obj}) \end{array} \right) \end{array}$$

Adding this to f-structure results in the structure at (143). The SUBJ and OBJ grammatical functions can be assigned to the PRED values ‘Jan’ and ‘Piet’ respectively, but this leaves an f-structure which is either incoherent or inconsistent, because it contains an ungoverned grammatical function. The optional XCOMP allows the possibility for a further PRED value to be added, thus avoiding ungrammaticality. ‘Marie’ does not specify the required grammatical functions to allow it to provide the XCOMP function. Accordingly, an unspecified f-structure is introduced that can provide the XCOMP and allow for a subsequent PRED value to govern the f-structure with PRED ‘Marie’.

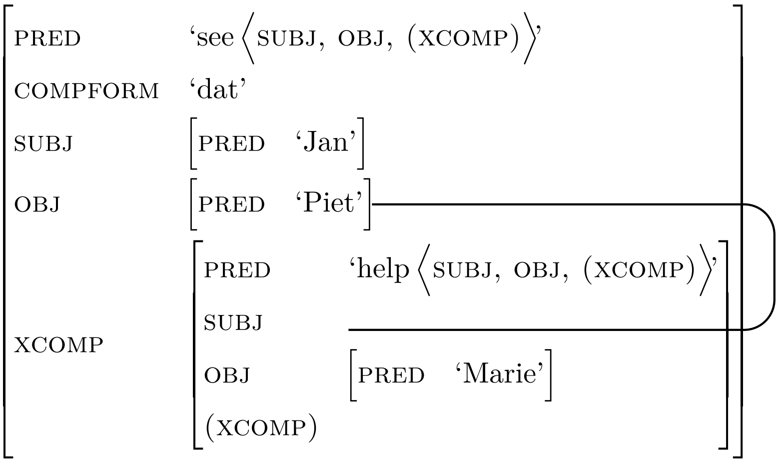
(143)
$$\begin{bmatrix} \textsc{pred} & \text{‘see}\left\langle \textsc{subj},\ \textsc{obj},\ (\textsc{xcomp}) \right\rangle\text{’} \\ \textsc{compform} & \text{‘dat’} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{‘Jan’}\end{bmatrix} \\ \textsc{obj} & \begin{bmatrix}\textsc{pred} & \text{‘Piet’}\end{bmatrix} \\ \textsc{xcomp} & \begin{bmatrix} \boxed{\textsc{pred}} & \\ \textsc{subj} & \\ \textsc{gf} & \begin{bmatrix}\textsc{pred} & \text{‘Marie’}\end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The sixth word of the fragment, *helpen* ‘help.INF’ has a similar lexical specification to that shown for *zag* in (142), and adding it produces the f-structure in (144).

---

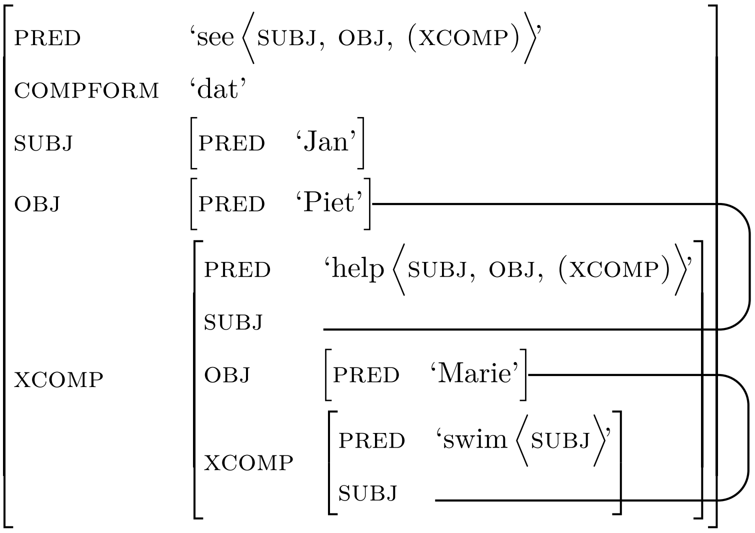
[26] I have amended the specification of selected grammatical functions for *zag*, which was given by Bresnan et al. (1982) as ⟨(↑ SUBJ)(↑ OBJ)(↑ VCOMP)⟩, to bring it in line with the assumptions of this thesis, but this change does not make a material difference to the analysis.

(144)



Because the XCOMP dependent of *helpen* is optional, the f-structure in (144) is complete and coherent. However, it is also possible to add the final word of the fragment, *zwemmen* 'swim.INF', resulting in the f-structure in (145) which corresponds to that given by Bresnan et al.

(145)



3.3.3.2 The principle of completeness

Completeness of an f-structure can only be assessed when a PRED value is present, so that selectional restrictions are available. Once a PRED value is present, so that the governed grammatical functions and any ordering restrictions are known, it is possible for grammatical functions to be assigned to dependents whose GF is underspecified, and for there to be some degree of prediction of any following grammatical functions.

Asudeh's assumptions for a processing model state that incremental production and parsing attempt to construct locally well-formed structures (Section 3.2.2, example 68b). Combining this assumption with the impact of a PRED being available allows two predictions to be made.

(146) **Prediction 2** Identical input strings may differ in their processing depending on whether or not a head PRED has already been contributed by an element in the string.

(147) **Prediction 3** In an incrementally growing f-structure that contains at least one PRED value that selects governable grammatical functions, a dependent that has multiple possibilities for attachment will be preferentially assigned a governed grammatical function rather than a non-governed GF.

3.3.3.3 The principle of consistency

The principle of consistency requires that there is only one value for each attribute of an f-structure. For dependents that are an argument of a governing PRED value, consistency of the f-structure can be computed as each dependent is introduced. However, where dependents precede a PRED value, the ability to assess well-formedness depends on whether grammatical functions are assigned on the basis of lexical specification or structurally, on the basis of ordering constraints in relation to the head.

Thus in Korean, the first two words of the complex phrase in (148) each introduce the constraint (SUBJ ↑). Even though a governing PRED value is not present, the principle of consistency is breached by two constituents bearing the same grammatical function, and so *nayka* can be identified as the left boundary of an embedded clause.

(148)

Tom-i	*nayka*	*etten*	*cilmwun-ul*	*cwu-nun-ci*	*a-nun*	*haksayng*
Tom-SBJ	1SG.SBJ	which	question-OBJ	give-PRS-COMP	know-ADN	student

"the student who Tom knows which question I gave" (Han, 2013, ex.11, p. 322)[27]

Thus the theory predicts that differences will be observed between cases where the grammatical function of a word is fully specified lexically, and cases where grammatical function is specified structurally in relation to a head PRED (149). This may be seen as a typological difference between languages, but may also occur when a language has some words for which the grammatical function is fully specified, and others for which the specification is partial.

(149) **Prediction 4** Before a governing PRED value is encountered, strings of dependents that include elements with lexically-specified grammatical functions will behave differently to strings where grammatical function is structurally assigned.

3.3.4 Summary of syntactic constraints

In Section 3.3 I have shown how the constraints and principles set out in LFG operate incrementally. Although each constraint is specified within a particular level of representation of the theory, the levels mutually constrain each other and so there is interaction of phrase-structure and functional constraints during incremental structure building.

---

[27] The original gloss has been amended for consistency with the approach to Korean nominal particles used in this thesis.

Notwithstanding these constraints, at any given point there may be multiple possibilities for the addition of a new word into the structure. These are points of ambiguity: the theory predicts that speakers of a language should be able to recover multiple partial interpretations at this point. Where speakers of a language show a preference for a particular interpretation at one of these points, hypotheses can be constructed on the basis of frequency, context, cognitive constraints etc. for empirical testing. I return to this further in Section 3.5.

3.3.5 Incremental computation of unbounded dependencies

An incremental approach to the analysis of an unbounded dependency is dependent on the point in an utterance where it is clear that such a dependency is present. For English this may be an explicit lexical cue such as a pronoun functioning as a question word, or a relative pronoun, or it may be a structural cue such as two nominal phrases in direct succession, such as where a fronted topic precedes a subject, or a reduced relative clause, where the first element is a nominal phrase and the clause an immediately preceding nominal phrase. Whatever the cue, the long-distance dependency is resolved later in the utterance after the left edge of the embedded clause is signalled. For Korean, there are constraints on a fronted topic if it occupies an argument position in an embedded clause, but there is no requirement for a fronted topic to result in a long-distance dependency, and so the constraints may be better seen as anaphoric rather than syntactic (Sohn, 1994). Korean relative clauses have no relative pronoun, and because all dependents of a head in Korean are syntactically optional, the first clear indication of a relative clause is a verb form carrying an adnominal inflection. Once the head that this adnominal modifies has been processed, there is a search over the existing syntactic and discourse context to resolve the dependency, similar to the resolution of backward anaphora.

A number of accounts of unbounded dependencies in LFG include discussion of linear precedence constraints in dependency resolution (Bresnan, 1995; Dalrymple et al., 2001; Dalrymple and King, 2013; Nadathur, 2013), which are directly relevant to an incremental theory. I adopt the notion of the syntactic anchor of an unbounded dependency, following Dalrymple and King (2013) and Nadathur (2013), to represent the predicate that governs the grammatical function whose value is shared with the relative pronoun or fronted topic. I propose that, once an unbounded dependency is opened, all PRED values that satisfy the language-specific path constraint and that select a grammatical function where a shared structure is possible, are provisionally assigned as the anchor at the point where the grammatical function could be filled. However, if subsequent material from the string can also be assigned that grammatical function, it replaces the provisional assignment of the unbounded dependency and forward search continues.

3.4 Building structure incrementally in Korean

3.4.1 Korean rules and constraints

For the following discussion of Korean, I use the LFG-based phrase structure rules, lexical specifications and morphosyntactic constraints provided by Cho and Sells (1995) together with further grammatical description from Sohn (1994) and Kiaer (2011).

Korean is a strongly head-final language where all arguments and modifiers to a head must precede it. Arguments may be omitted if they can be inferred from the context. However, if an argument or a modifier to a head has been expressed, then its head must also be expressed. Embedding of phrases and clauses within higher phrases is permitted, and there is free ordering of the arguments and modifiers within the phrase. However, with very few exceptions[28] scrambling out of the phrase is restricted to one constituent which can be extracted from an arbitrary depth to the front of the sentence, and which is marked with the particle *-un/-nun*, often referred to as the topic particle (Sohn, 1994).

Korean government guidelines on grammar teaching describe nine word classes: nouns, pronouns, numerals, verbs of action, verbs of description, adnominals, adverbs, exclamations, and particles (Ihm et al., 2001). Only nouns and verbs are open word classes, other word classes are closed.

The grammar that I use to derive the set of strategies and conditions builds on the proposal of Cho and Sells (1995), who assume two major word categories, N and V, which have single phrasal projections, N' and V'. Attachment of phrasal arguments and adjuncts takes place at X', and X' nodes are binary-branching. Other categories are non-projecting and attach at X^0. Within this very general schema, ordering constraints are accounted for by the notion of TYPE, which is a property of words and inflectional morphemes, and which introduces lexical constraints on c-structure and f-structure.

Words of all categories are lexically specified to have a type value, which may be V-SIS, N-SIS, COORD or NO, which is derived by a morphological process[29]. The X' nodes projected by a terminal node inherit the type value of that terminal node. A phrase of type V-SIS must attach at V' and a phrase of type N-SIS must attach at N'. Words from non-projecting categories of types V-SIS and N-SIS attach at V^0 and N^0 respectively. Type COORD is carried by words inflected with a coordinating particle (e.g. *kwa, hako*), and type NO is carried by words inflected with sentence-final particles (e.g. *-yo* 'POL', *-supnita* 'FORMAL.INDIC'). This constraint is similar in nature to the constraints on sister nodes described by Nordlinger (1998). Following Dalrymple (2001), I use the variable $*\!\succ$ to represent a node's immediate right sister. The formal definition of $*\!\succ$ is given in (150).

---

[28] See Kempson and Kiaer (2010) for prosodically-licensed examples where two sister-arguments are fronted, or where an element of the sentence is scrambled to the right of the matrix verb for emphasis.

[29] Cho and Sells (1995) also propose the type value ∼TYPE, but this is available only to morphemes that cannot be introduced into the syntax, and so it is not considered here.

(150) $$*\rangle \ = n \quad \Rightarrow \quad \begin{aligned} & R = \{r : M(r) = M(*) \ \land \ * \prec r\} \land \\ & n \in R \land \\ & \neg\exists y. y \in R \ \land \ y \prec n \end{aligned}$$

> The immediate right sister $*\rangle$ of node $*$ is n if n is a member of the set R of right sisters of $*$, and there is no member of R that precedes n.

Cho and Sells' type definitions are not specified formally, but are presented with "intended interpretations" of the type values (p. 165). Here I provide formal definitions of the four values of TYPE that are available to words: N-SIS (151), V-SIS (152), COORD (153), and NO (154). In the following definitions, Cho and Sells' description of TYPE in relation to a constituent X annotated with a type value is followed by a formal definition in relation to the c-structure node $*$ that corresponds to X. I follow Dalrymple (2017) in defining category as a feature matrix, with N being an abbreviation for $[\text{N} +, \text{V} -]$ and V an abbreviation for $[\text{N} -, \text{V} +]$.

(151) "N-SIS: if X is an immediate left sister of a constituent C, C is a projection of N" (Cho and Sells, 1995, p. 165, ex. (79b))

$$\underset{\text{TYPE:N-SIS}}{*} \ \Rightarrow \ \lambda(*\rangle\ ) = [\text{N} +, \text{V} -]$$

(152) "V-SIS: if X is an immediate left sister of a constituent C, C is a projection of V" (*ibid.* p. 165, ex. (79a))

$$\underset{\text{TYPE:V-SIS}}{*} \ \Rightarrow \ \lambda(*\rangle\ ) = [\text{N} -, \text{V} +]$$

(153) "COORD: in an n-ary branching structure, every sister of X matches X in terms of category and bar level" (*ibid.* p. 165, ex. (79c))

$$\underset{\text{TYPE:COORD}}{*} \ \Rightarrow \ \forall n.(M(n) = M(*)) \ \Rightarrow \ \lambda(n) = \lambda(*)$$

(154) "NO: X has no right sister" (*ibid.* p. 165, ex. (79e))

$$\underset{\text{TYPE:NO}}{*} \ \Rightarrow \ \neg *\rangle$$

Cho and Sells further propose that when a node is a left sister of another node, its type annotation is associated with an f-structure condition as shown in (155). Where a node is not a left sister of another node, case (155b) applies.

(155) a. TYPE:X-SIS $\Rightarrow$ $(\uparrow \text{GF}) = \downarrow$

b. otherwise $\uparrow = \downarrow$

Adding the type specifications N-SIS, V-SIS, and NO to the overall phrase structure schema, together with the f-structure conditions from (155), gives the phrase structure rules in (156)–(159), where $\widehat{\text{W}}$ stands for a non-projecting category[30].

(156) $$\text{N}' \rightarrow \left\{ \begin{array}{c} \text{N}^0 \\ (\hat{*}\ \text{TYPE}) = \text{TYPE} \\ \uparrow=\downarrow \end{array} \;\middle|\; \left\{ \begin{array}{c} \text{N}' \\ \text{TYPE:N-SIS} \\ (\uparrow \text{GF}) =\downarrow \end{array} \;\middle|\; \begin{array}{c} \text{V}' \\ \text{TYPE:N-SIS} \\ (\uparrow \text{GF}) =\downarrow \end{array} \right\} \begin{array}{c} \text{N}' \\ (\hat{*}\ \text{TYPE}) = \text{TYPE} \\ \uparrow=\downarrow \end{array} \right\}$$

(157) $$\text{V}' \rightarrow \left\{ \begin{array}{c} \text{V}^0 \\ (\hat{*}\ \text{TYPE}) = \text{TYPE} \\ \uparrow=\downarrow \end{array} \;\middle|\; \left\{ \begin{array}{c} \text{N}' \\ \text{TYPE:V-SIS} \\ (\uparrow \text{GF}) =\downarrow \end{array} \;\middle|\; \begin{array}{c} \text{V}' \\ \text{TYPE:V-SIS} \\ (\uparrow \text{GF}) =\downarrow \end{array} \right\} \begin{array}{c} \text{V}' \\ (\hat{*}\ \text{TYPE}) = \text{TYPE} \\ \uparrow=\downarrow \end{array} \right\}$$

(158) $$\text{N}^0 \rightarrow \begin{array}{c} \widehat{\text{W}} \\ \text{TYPE:N-SIS} \\ (\uparrow \text{GF}) = \downarrow \end{array} \quad \begin{array}{c} \text{N}^0 \\ (\hat{*}\ \text{TYPE}) = \text{TYPE} \\ \uparrow=\downarrow \end{array}$$

(159) $$\text{V}^0 \rightarrow \begin{array}{c} \widehat{\text{W}} \\ \text{TYPE:V-SIS} \\ (\uparrow \text{GF}) = \downarrow \end{array} \quad \begin{array}{c} \text{V}^0 \\ (\hat{*}\ \text{TYPE}) = \text{TYPE} \\ \uparrow=\downarrow \end{array}$$

Thus a word of a given category selects daughters with the corresponding type, and a word of a given type constrains its mother to be of the corresponding category. Accordingly, in constructing the mother and daughter sets for Korean I will construct the daughter sets for the two categories N and V, and the mother sets for the two types N:SIS and V:SIS using a category variable which can be instantiated by N, V, or $\widehat{\text{W}}$. This captures the inability of non-projecting categories to have attaching daughters. The sets derived from the above phrase-structure rules are given in (160)–(161).

(160) Daughter sets by category.

a. $$\text{N}^d_{d\prec\text{N}} : \left\{ \begin{array}{c} \text{N} \\ \text{TYPE:N-SIS} \end{array} , \begin{array}{c} \text{V} \\ \text{TYPE:N-SIS} \end{array} , \begin{array}{c} \widehat{\text{W}} \\ \text{TYPE:N-SIS} \end{array} \right\}$$

b. $\text{N}^d_{\text{N}\prec d} : \{\ \emptyset\ \}$

c. $$\text{V}^d_{d\prec\text{V}} : \left\{ \begin{array}{c} \text{N} \\ \text{TYPE:V-SIS} \end{array} , \begin{array}{c} \text{V} \\ \text{TYPE:V-SIS} \\ \uparrow=\downarrow \end{array} , \begin{array}{c} \widehat{\text{W}} \\ \text{TYPE:V-SIS} \end{array} \right\}$$

d. $\text{V}^d_{\text{V}\prec d} : \{\ \emptyset\ \}$

(161) Mother sets by type.

a. $$\underset{\text{TYPE:N-SIS}}{\text{X}^m_{m\prec\text{X}}} : \{\ \emptyset\ \}$$

b. $$\underset{\text{TYPE:N-SIS}}{\text{X}^m_{\text{X}\prec m}} : \{\ \text{N}\ \}$$

---

[30] The coordination rules from Cho and Sells (1995) are left for future work.

c. $\underset{\text{TYPE:V-SIS}}{\mathrm{X}^{m}_{m \prec \mathrm{X}}}$: { ∅ }

d. $\underset{\text{TYPE:V-SIS}}{\mathrm{X}^{m}_{\mathrm{X} \prec m}}$: { V }

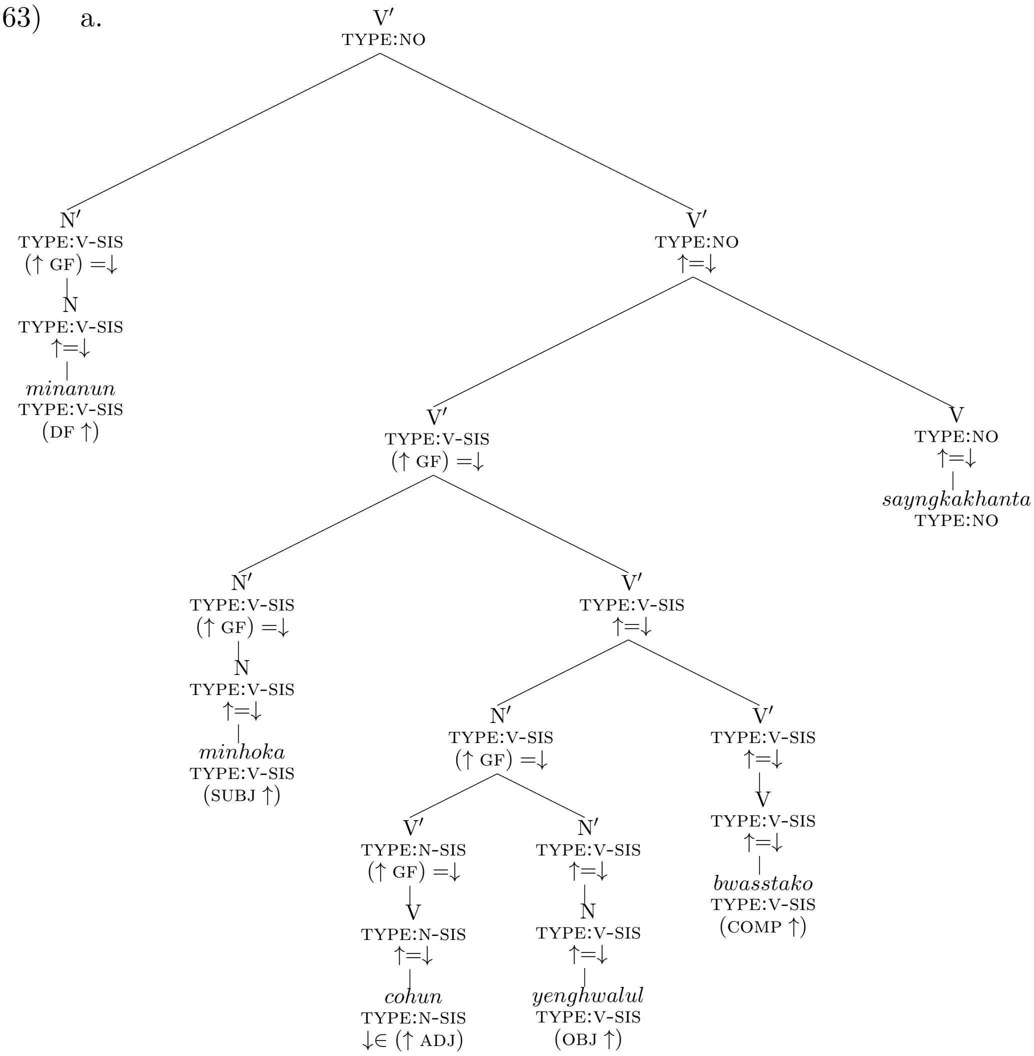
Note that the succeeding daughter sets (160b) and (160d), and the preceding mother sets (161a) (161c) are the empty set: this reflects the strongly head-final nature of Korean.

3.4.2 Worked example

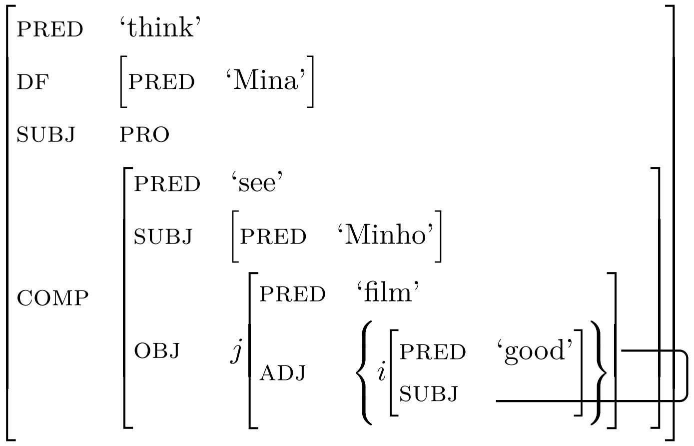
I now work through an example of incremental structure-building in Korean using the example sentence in (162), which has the target c-structure and f-structure shown in (163). The example demonstrates how the general theory can be applied, but also highlights questions of non-structural factors that come into play when a parsing choice point is reached. The mother-daughter set combinations for attachment as a succeeding daughter, where both sets are the empty set, are ignored.

(162) *mina-nun minho-ka coh-un yenghwa-lul bwass-tako sayngkakha-nta*
Mina-DF Minho-SBJ good-ADN film-OBJ saw-COMP think-PRS.PLAIN
"As for Mina, she thinks that Minho saw a good film."

(163) a.



b.



I assume the lexical specifications in Table 3.6.

At the start of the sentence there is no structure, and so rule (157) is instantiated with information from the lexical entry for *minanun* 'Mina-DF'. Because the first word contributes the grammatical function specification (DF ↑), an underspecified f-structure

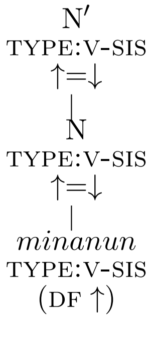
Table 3.6: Lexical specifications assumed for the anslysis of (162)

bwasstako	V	[TYPE:V-SIS] (↑ PRED) = ‘see ⟨SUBJ, OBJ⟩’ (COMP ↑)
cohun	V	[TYPE:N-SIS] (↑ PRED) = ‘good ⟨SUBJ⟩’ (↓ SUBJ) = ↑ ↓ ∈ (↑ ADJ)
minanun	N	[TYPE:V-SIS] (↑ PRED) = ‘Mina’ (DF ↑)
minhoka	N	[TYPE:V-SIS] (↑ PRED) = ‘Minho’ (SUBJ ↑)
sayngkakhanta	V	[TYPE:NO] (↑ PRED) = ‘think ⟨SUBJ, COMP⟩’
yenghwalul	N	[TYPE:V-SIS] (↑ PRED) = ‘film’ (COMP ↑)

is introduced, but there is not yet an underspecified node as the initial phrase does not require attachment. The outcome is shown in (164).

(164) *mina-nun …*
Mina-DF …
“As for Mina, …”

C_1:



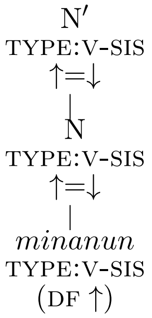
$$g\begin{bmatrix}\text{PRED} & \bullet \\ \text{DF} & f\begin{bmatrix}\text{PRED} & \text{‘Mina’}\end{bmatrix}\end{bmatrix}$$

The next word, *minhoka* ‘Minho-SBJ’ has category N and type V-SIS, producing the string in (165). The relevant succeeding mother (161d) and preceding daughter sets (160a) do not allow for attachment of C_1 as the left daughter of C2. Accordingly an uninstantiated node is introduced, which is constrained by the type value of its two daughters to be projected by a word whose category has the feature [CAT V +]. The process is shown in (166).

(165) *mina-nun minho-ka* ...
mina-DF minho-SBJ ...

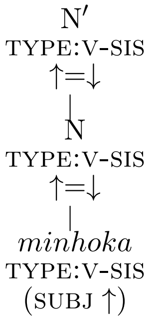
"As for Mina, Minho ..."

(166) a. C$_1$:



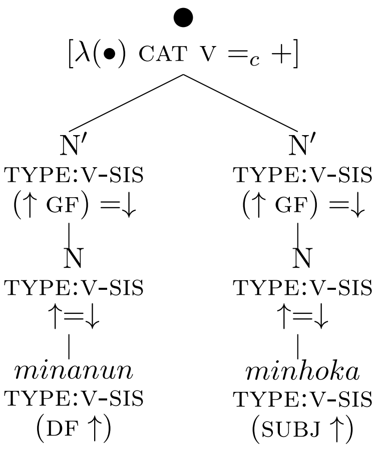
$$g\begin{bmatrix}\textsc{pred} & \bullet \\ \textsc{df} & f\begin{bmatrix}\textsc{pred} & \text{'Mina'}\end{bmatrix}\end{bmatrix}$$

b. + C$'_2$:



$$i\begin{bmatrix}\textsc{pred} & \bullet \\ \textsc{subj} & h\begin{bmatrix}\textsc{pred} & \text{'Minho'}\end{bmatrix}\end{bmatrix}$$

c. = C$_2$:

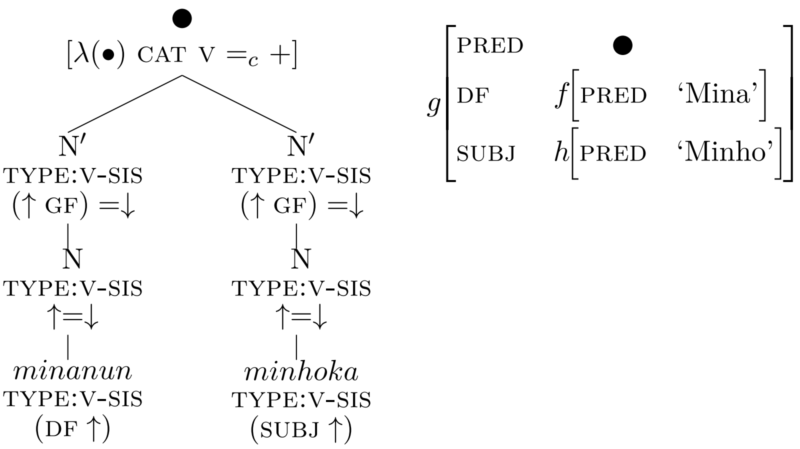


$$g\begin{bmatrix}\textsc{pred} & \bullet \\ \textsc{df} & f\begin{bmatrix}\textsc{pred} & \text{'Mina'}\end{bmatrix} \\ \textsc{subj} & h\begin{bmatrix}\textsc{pred} & \text{'Minho'}\end{bmatrix}\end{bmatrix}$$

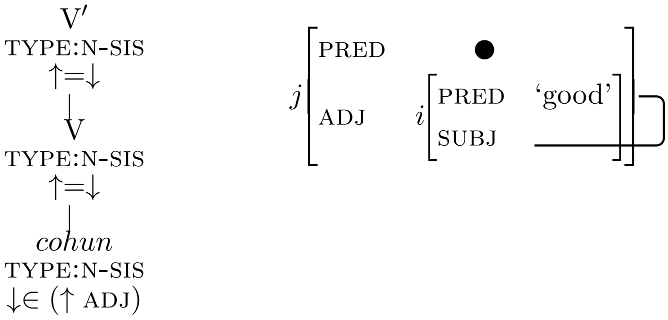
The third word, *cohun* has category V and type N:SIS. Its lexical specification also introduces functional constraints: as an adjectival verb it selects only a SUBJ grammatical function, and because it carries the adnominal marker *-un*, it functions as an attributive adjective whose SUBJ is coreferent with the the noun it modifies (Yeon and Brown, 2011). The string and starting position for attachment are shown in (167).

(167) *mina-nun minho-ka coh-un...*
Mina-DF Minho-SBJ good-ADN ...
"As for Mina, Minho ...a good ..."

C$_2$:



C′$_3$:

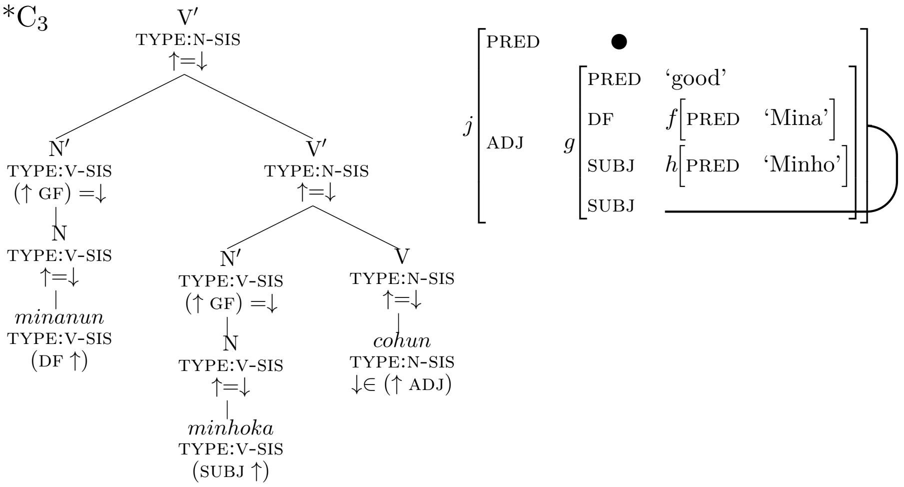


The c-structure C′$_3$ has category V, and thus the category specification [N − V +]. Also, its left daughter set (160c) contains N, and so in terms of its type constraints it could instantiate the underspecified node. However, the functional constraint on its subject precludes this because the resulting f-structure would be incoherent, with the SUBJ value provided both by f-structure *h* and f-structure *j*[31]. This constraint also operates to prevent attachment of C′ within the right edge of C$_2$ between the topmost node and the N′ node projected by *minhoka*. The two ungrammatical attachment possibilities are shown in (168). Because the Korean phrase structure rules are binary branching, the instantiation of the underspecified node in (168a) results in an additional V′ being added to provide an attachment site for the leftmost N′ phrase, above the V′ already present as part of C′$_3$.

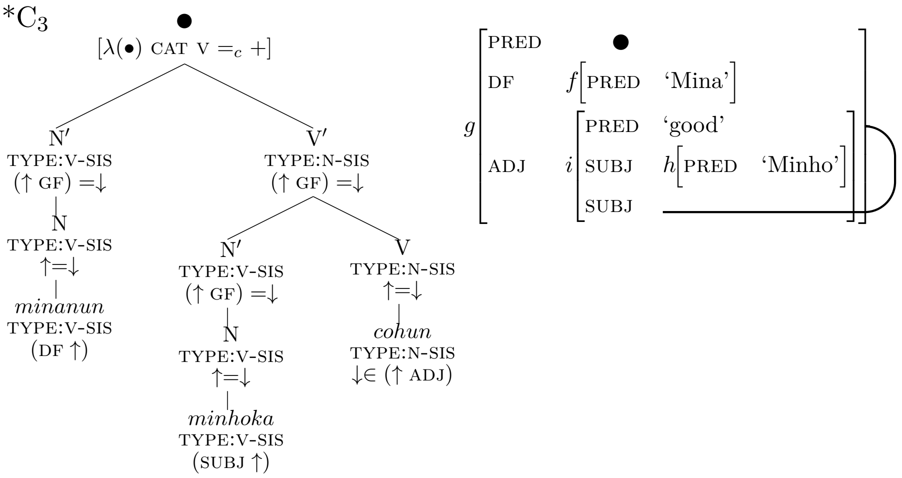
---

[31] Although Korean does permit multiple instances of SUBJ or OBJ with the same predicate (Sohn, 1994) this is semantically constrained (Ryu, 2014). I assume that the syntactic constraints apply initially with any semantic evaluation and repair of the representation taking place later during processing.

(168) a. *C$_3$

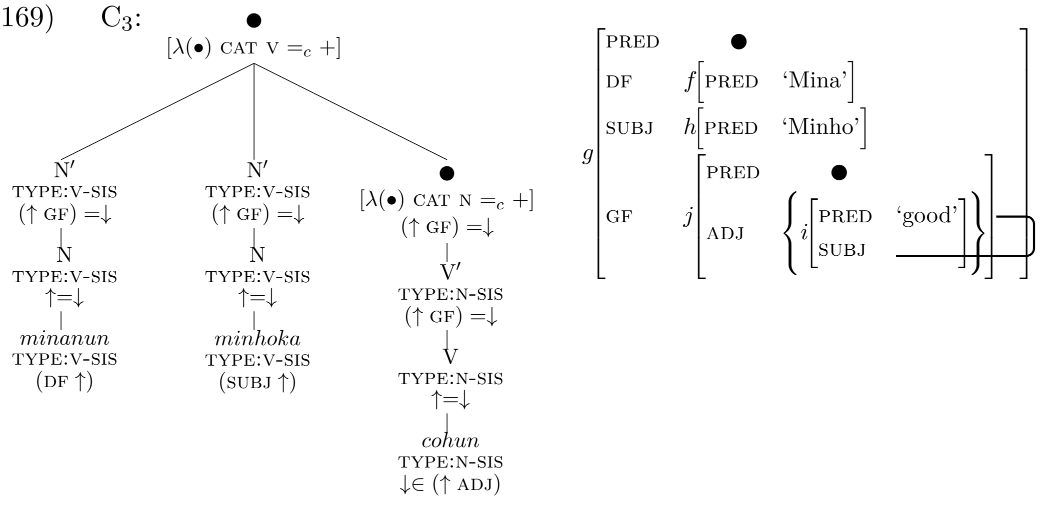


b. *C$_3$



Because of these constraints, C′$_3$ must be added to the structure as the right daughter of the underspecified node: right daughters of the N′ phrase projected by *minhoka* are not allowed. Adding directly below the underspecified node would add the constraint [λ(•) CAT N $=_c$ +] to the existing [λ(•) CAT V $=_c$ +] constraint on the node. As categories N and V are fully specified as [N+ V−] and [N− V+] respectively, this would make the node impossible to instantiate. Accordingly, a further underspecified node is added below the root node, to allow C′$_3$ to attach. This results in the situation shown in (169).

(169) C$_3$:

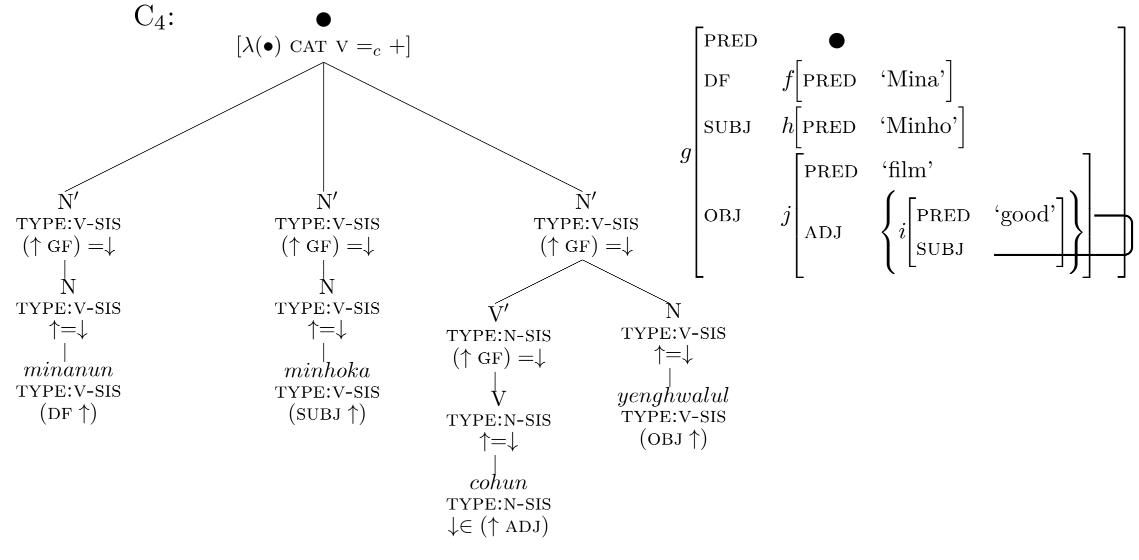


The fourth word, *yenghwa-lul* ‘film-OBJ’ has category N and type V-SIS, and so can instantiate the lower underspecified node. This results in the c-/f-structure pair in (170).

(170) *mina-nun* *minho-ka* *coh-un* *yenghwa-lul* ...
Mina-DF Minho-SBJ good-ADN film-OBJ ...

“As for Mina, Minho ...a good film ...”

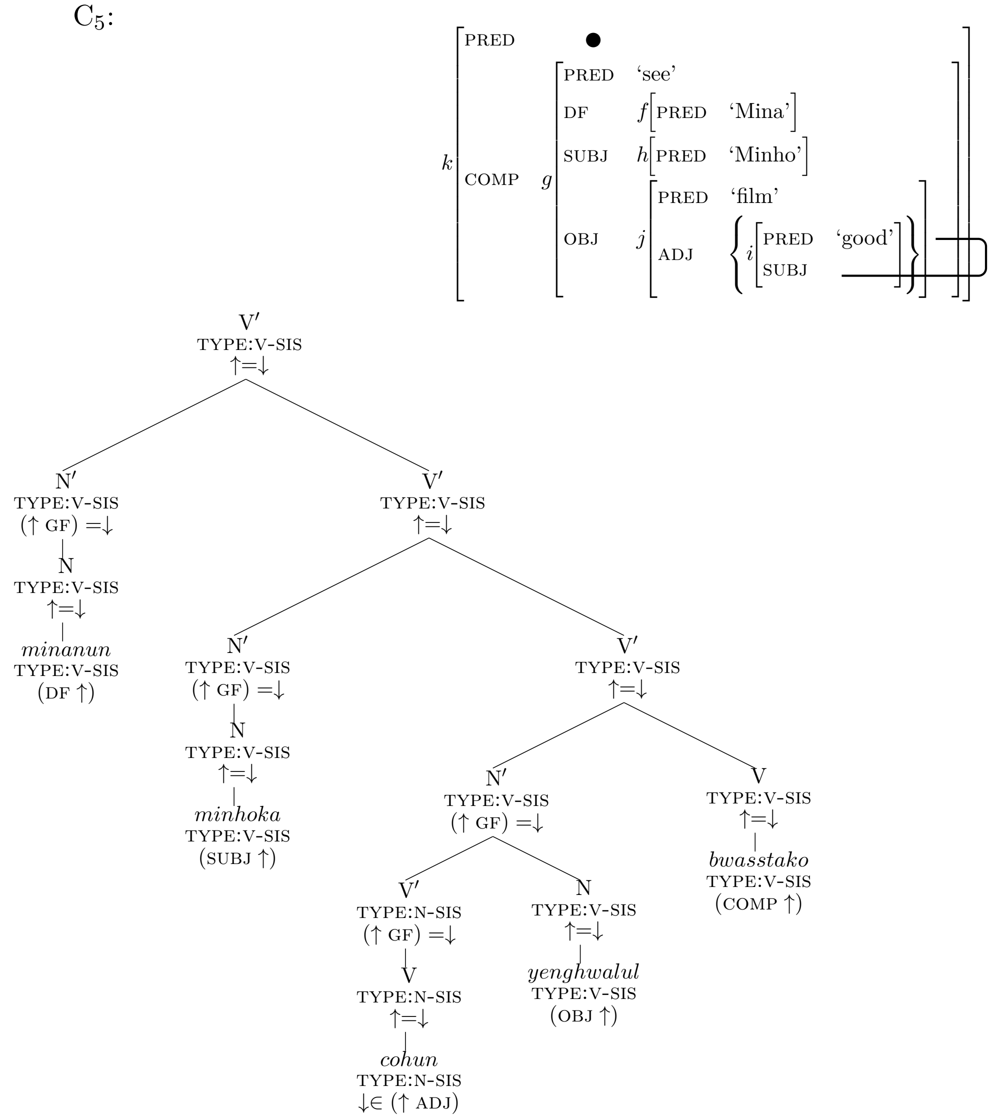
C$_4$:



The fifth word, *bwasstako* 'saw-COMP' has category V, type V-SIS. It selects SUBJ and OBJ grammatical functions and carries the f-structure constraint (COMP ↑). Its category is compatible with the constraint on the underspecified node, and its valency is compatible with the grammatical functions already present in f-structure g, and so it could instantiate the underspecified node as shown in (171). Similarly to the instantiation in (168), this requires additional V′ nodes above the first V′ projected by *bwasstako*.

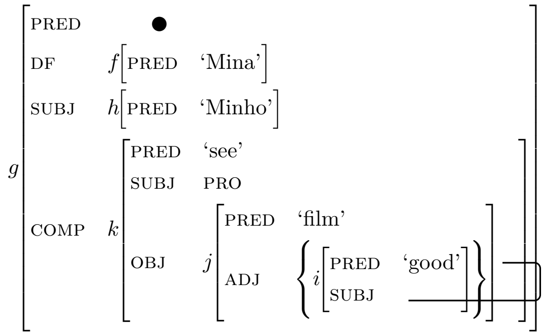
(171) *mina-nun minho-ka coh-un yenghwa-lul bwass-tako* ...
Mina-DF Minho-SBJ good-ADN film-OBJ saw-COMP ...
"As for Mina, ...that Minho saw a good film."

C$_5$:

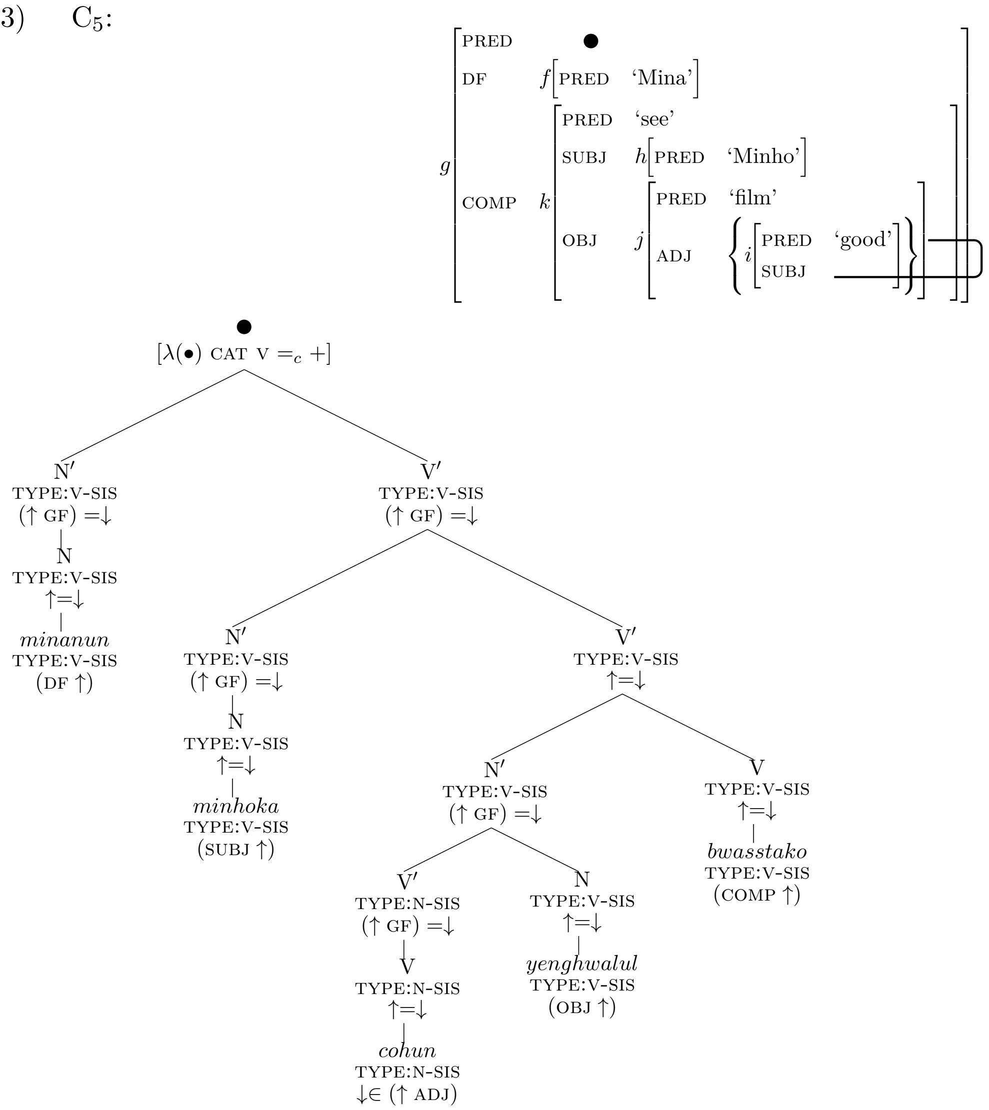


An alternative attachment possibility is to attach *bwasstako* within the right edge, between the underspecified node and one or more of its daughters, starting with the rightmost daughter. Example (172) shows just the f-structure for attachment of *bwasstako* between the underspecified node and its rightmost daughter. Example (173) shows the full analsysis if both the second and third daughters of the underspecified node become daughters of the phrase projected by *bwasstako*.

(172)



(173) C_5:



Given the syntactically-licensed alternatives, the question arises as to how the choice is made between them. I return to this in Section 3.5.

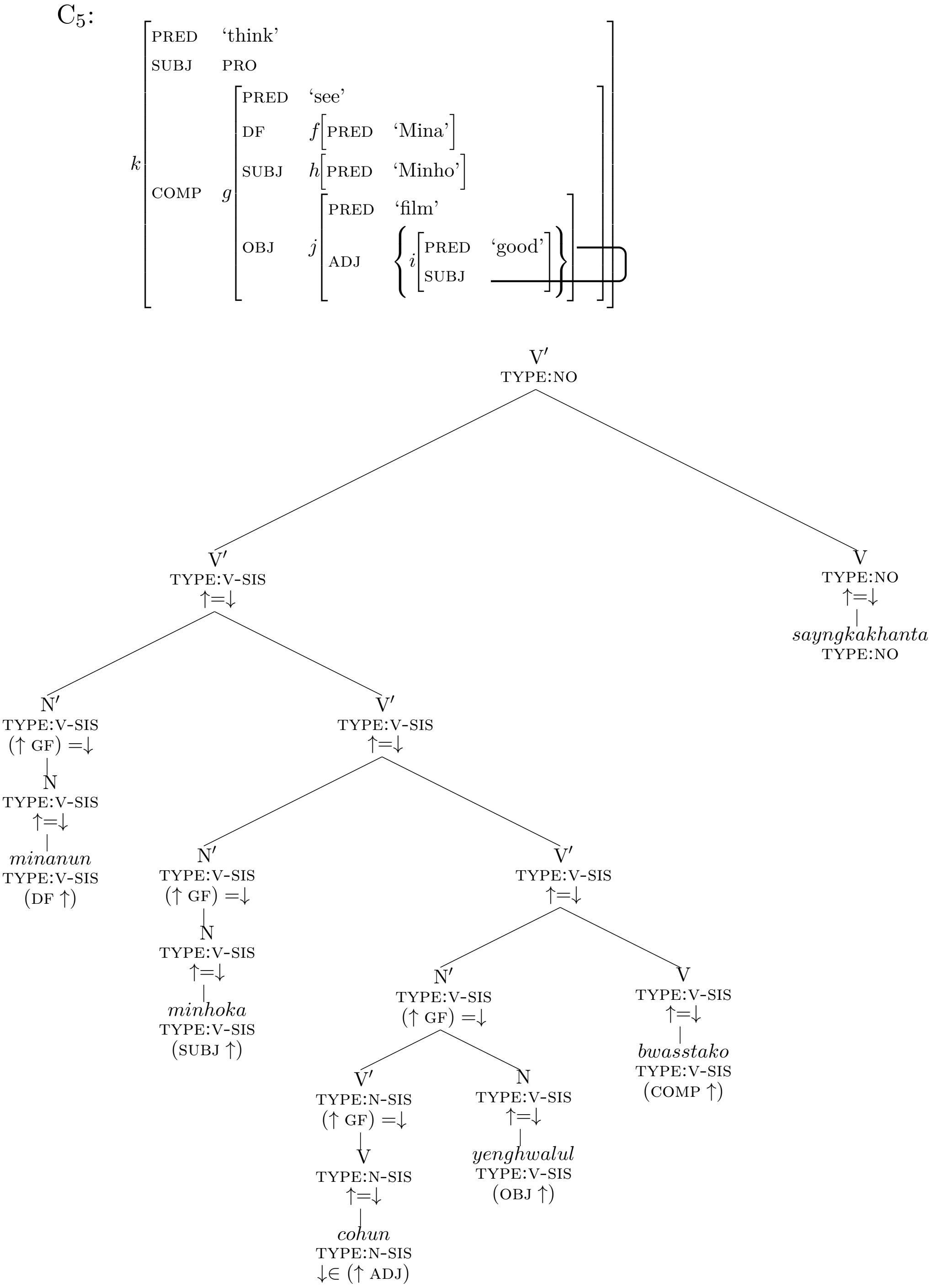
The sixth word, *sayngkakha-nta* ‘think-PRS.PLAIN’, has category V and type NO, and selects SUBJ and COMP grammatical functions. Its type constrains it to be the topmost node of c-structure. If the attachment in (172) was chosen, the phrase projected by *sayngkakhanta* attaches above c-structure C_5, and if the attachment in (173) was chosen, category V is compatible with the constraint on the underspecified node, and so this node is instantiated with the V' node projected by *sayngkakhanta*. The c-/f-structure pairs for these two choices are given in (174) and (175), and the f-structure resulting from adding *sayngkakhanta* to (172) is given at (176).

(174)

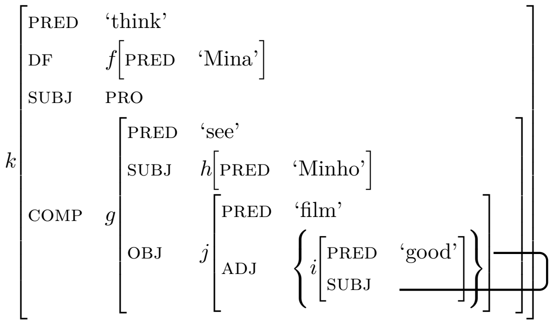
mina-nun	*minho-ka*	*coh-un*	*yenghwa-lul*	*bwass-tako*	*sayngkakha-nta*
Mina-DF	Minho-SBJ	good-ADN	film-OBJ	saw-COMP	think-PRS.PLAIN

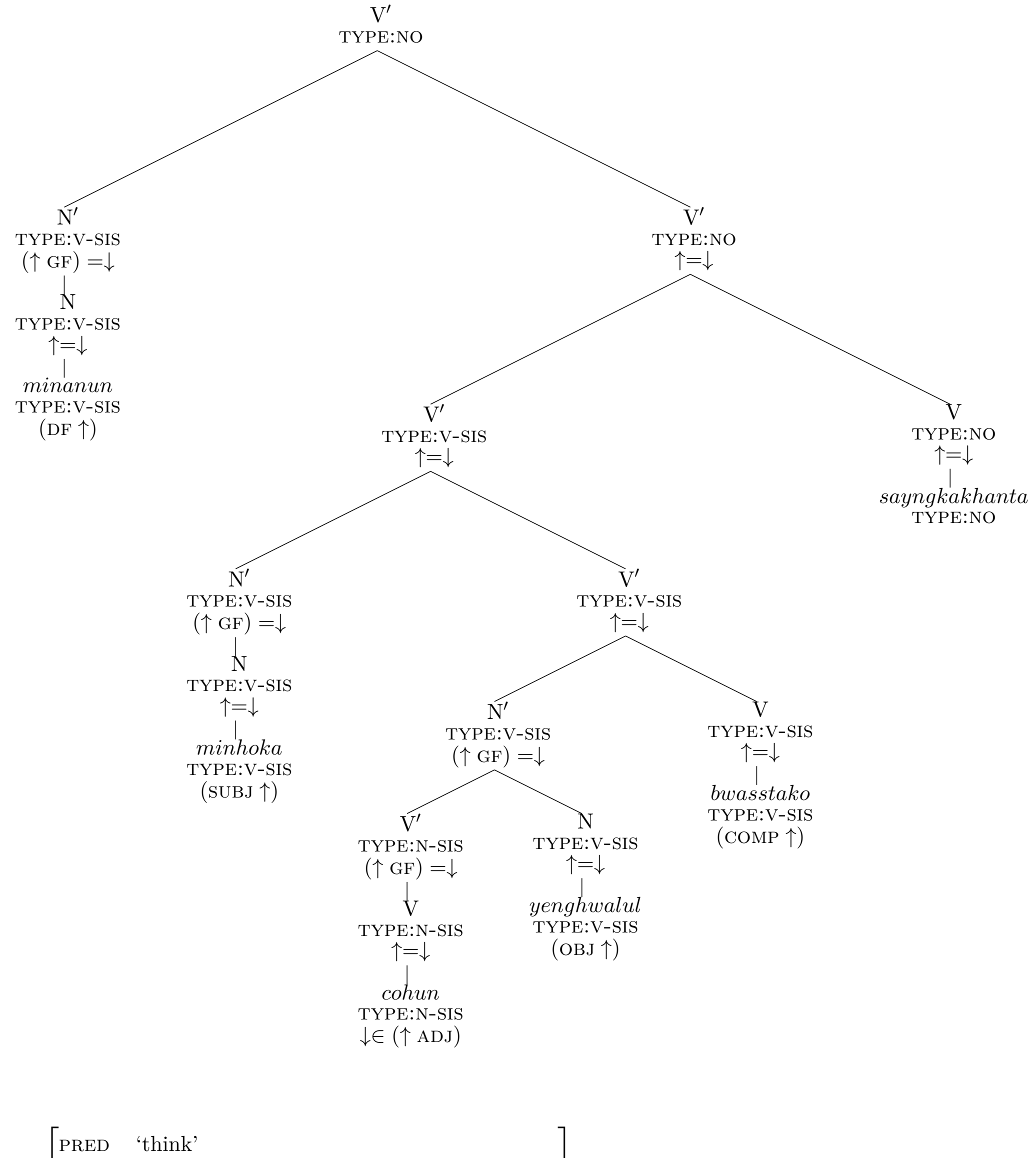
“As for Mina, she thinks that Minho saw a good film.”

C$_5$:

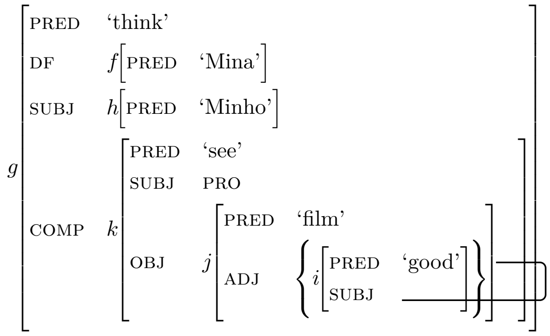


(175) C_5:


k
PRED 'think'
DF f [PRED 'Mina']
SUBJ PRO
COMP g
PRED 'see'
SUBJ h [PRED 'Minho']
OBJ j
PRED 'film'
ADJ { i [PRED 'good' SUBJ] }


V′
TYPE:NO
N′
TYPE:V-SIS
(↑ GF) =↓
N
TYPE:V-SIS
↑=↓
minanun
TYPE:V-SIS
(DF ↑)
V′
TYPE:NO
↑=↓
V′
TYPE:V-SIS
↑=↓
N′
TYPE:V-SIS
(↑ GF) =↓
N
TYPE:V-SIS
↑=↓
minhoka
TYPE:V-SIS
(SUBJ ↑)
V′
TYPE:V-SIS
↑=↓
N′
TYPE:V-SIS
(↑ GF) =↓
V′
TYPE:N-SIS
(↑ GF) =↓
V
TYPE:N-SIS
↑=↓
cohun
TYPE:N-SIS
↓∈ (↑ ADJ)
N
TYPE:V-SIS
↑=↓
yenghwalul
TYPE:V-SIS
(OBJ ↑)
V
TYPE:V-SIS
↑=↓
bwasstako
TYPE:V-SIS
(COMP ↑)
V
TYPE:NO
↑=↓
sayngkakhanta
TYPE:NO

(176)


g
PRED 'think'
DF f [PRED 'Mina']
SUBJ h [PRED 'Minho']
COMP k
PRED 'see'
SUBJ PRO
OBJ j
PRED 'film'
ADJ { i [PRED 'good' SUBJ] }

There are subtle differences between the readings obtained from the three structures. The most straightforward reading is (175), which encourages the matrix DF *mina* to be interpreted also as the matrix SUBJ, and which is also the intended reading of the sentence.

3.5 Choosing between structural growth options

In Chapter 2, I discussed several proposals regarding the factors that influence which choice is made when a new word is introduced to an incrementally growing utterance. These included the Minimal Chain and Late Closure strategies (Frazier and Fodor, 1978), lexically-determined semantic preferences (Ford et al., 1982), contextually-determined structural and semantic preferences (MacDonald et al., 1994; Hale, 2001; Levy, 2008), and interactions between prosody and syntax (Fodor, 2002, 2013). In this section I review the constraints on structural growth introduced by the incremental theory of LFG in relation to the above proposals. I will focus in particular on the role of functional constraints and f-structure growth, as these aspects distinguish LFG from other theories.

The Minimal Attachment and Late Closure strategies are perhaps the longest-standing proposal regarding constraints on building syntactic structure. Minimal Attachment is reflected in my assumption that only the minimal necessary structure is added incrementally to allow a grammatical attachment to take place. The notion of an underspecified node to c-structure removes the need to posit whole phrases, which might then require further revision (e.g. adding an IP when a CP was required). However, additional structure can also be added as underspecified f-structures, either projected from underspecified nodes or introduced lexically.

A consequence of the minimality constraint is that any underspecified f-structure must contain at least one feature-value pair. The principles of coherence and completeness interact with relationship between c-structure dominance and f-structure containment to require that information can only be added inside this underspecified f-structure until the principles of coherence and completeness are satisfied. In other words, when a new f-structure is added within an existing f-structure, the new f-structure must be completed first. This may entail further f-structures being added within the new f-structure. The principle derivable from the constraint is that, as f-structures are created, information is added from the inside outwards. This principle applies to languages that are configurational within the definition of this thesis: the question of how the principle might operate for languages with highly free word order is left for future work.

The universality of Frazier's Late Closure strategy was challenged by empirical data from English and other languages. The incremental theory of LFG allows for a preference for Late Closure to interact with lexical selection of governed grammatical functions. In particular, where the sentence context includes a predicate that governs grammatical functions, structurally ambiguous content takes a governed grammatical function in preference to an ADJ or an XADJ: the principle of completeness is preferentially satisfied. The theory of Ford et al. (1982) foreshadows this, although their experimental data concerned preferences between low and high attachment for modifiers, rather than arguments. It should be possible for the incremental theory to incorporate Ford et al.'s proposals on lexical preferences for modifier attachment, but again this is left for future work.

The incremental theory proposes that structure grows bottom-up (or inside-out): information on attachment constraints is lexically specified and there is no distinct knowledge of phrase-structure rules or application of a parsing algorithm in the sense of Aho and Ullman (1972). The theory can therefore be seen as more in the tradition of MacDonald et al. (1994) rather than Hale (2001) or Levy (2008). The initial treatment focuses on syntactic categorial and grammatical function constraints and does not yet consider other factors included in MacDonald et al. such as person, number, tense, aspect, mood, or the sentential semantic context.

Prosodic boundaries can be incorporated into the theory if the assumption is that they are associated with the left edge of a constituent, indicating syntactic closure in Frazier's or Ford et al.'s terms. An initial proposal is that, where an f-structure *may* still have lexical content added to it, a prosodic boundary indicates that no more lexical content *will* be added to it, and so final assessment of completeness, coherence and consistency can take place. This proposal is explored further in the model in Chapter 6.

Even with the above constraints framed in terms of incremental LFG, structure-building continues to have substantial and pervasive ambiguity. Thus the ambiguity in the Korean example in Section 3.4 remains after considering syntactic constraints. LFG allows for the incremental computation of semantic content and a discourse representation, alongside any inferences that may be made by non-linguistic processing based on sentence content and wider context. These questions, as well as the relationship between the theory and experimental evidence on the time-course and scope of reanalysis or repair are also left for future work.

3.6 Summary

In this chapter I introduced LFG as a modular, declarative, constraint-based theory of the relationship between linguistic form and meaning. I then built on the principles proposed by Asudeh (2012) to elaborate an incremental theory of LFG, considering syntactic constraints on word-by-word structure growth. The theory is rooted in the work of Frazier and Fodor (1978) and MacDonald et al. (1994) among others, but uses LFG's f-structure and grammatical functions to describe constraints and make specific predictions about how incremental structure-building will take place. I used a worked example of structure growth in Korean as an initial test of the theory, with the residual ambiguities in the analysis illustrating the boundaries of the theory and highlighting opportunities for future work.

I now move on to test the theory computationally using ACT-R, a computational architecture that aims to reflect human cognitive constraints. Constructing a model in ACT-R requires a hypothesis to be specified precisely, and fit between model performance and human data can be used to assess the plausibility of proposed accounts of human cognition. In Chapter 4, I introduce ACT-R and discuss published models of language processing in that, and other, computational cognitive architectures. I then continue in Chapter 5 to identify how the incremental theory of LFG can be adapted for the purposes of cognitive modelling, and present the results of modelling in Chapter 6.

Chapter 4

Cognitive modelling of language processing

In Chapter 2, I discussed the role that grammar plays in theories of language processing, and considered some of the processing phenomena that have been investigated empirically. In Chapter 3, I elaborated an incremental theory of LFG, which aims to show how linguistic knowledge and grammatical constraints apply incrementally during the comprehension of an unfolding utterance, and to allow the contribution of non-language-specialised factors to be described precisely in relation to the grammar. In the thesis thus far, grammar and processing have remained theoretically distinct, and in this Chapter I introduce computational modelling as a linking hypothesis between the two.

One group of linking hypotheses already introduced in Chapter 2 are complexity metrics, a linking hypothesis between theories of grammar, theories of language processing, and experimental measures of performance in language processing (e.g eye-tracking times, self-paced reading times) or of brain activity (measured by e.g. fMRI or EEG). However, as pointed out by VanWagenen et al. (2014), when calculating a complexity metric, a language modeller makes assumptions about the nature of the grammar, the processing algorithm, and the probability distribution of alternatives at any given choice point during the parse. Complexity metrics can be compared for different combinations of these assumptions, and can also be correlated with empirical measures of human processing. However, where correlations are made with empirical data, a further linking hypothesis is required about the assumed relationship between the two.

An alternative to complexity metrics is to use a cognitive architecture. Cognitive architectures are computational representations of theories of mind, which seek to replicate the cognitive capacity and constraints of the human brain. A particular cognitive architecture can then be used as the basis for a variety of models of cognitive tasks. If the modeller succeeds in simulating experimental findings, this can be taken as evidence in support of the general theory of mind as well as the specific process model assumed by the modeller. Well-established cognitive architectures are supported by a number of models across a broad range of cognitive tasks.

As the research field has become established, there have been efforts to identify commonalities between the assumptions of different cognitive architectures, with Laird et al. (2017) proposing a standard model of the mind across the architectures ACT-R (Anderson, 2007), Sigma (Rosenbloom et al., 2016), and Soar (Laird, 2012). This model, now termed the Common Model of Cognition, is being used as the basis to assess the assumptions inherent in attempts to simulate human language processing (e.g. Lindes, 2018).

The model developed in this thesis uses the ACT-R cognitive environment. In Section 4.1 I introduce the environment and outline how it operates, before reviewing other work in language processing in Section 4.2. The chapter concludes in Section 4.3 with a discussion of the theoretical and practical issues for an LFG-based model that arise from the work reviewed.

4.1 The ACT-R cognitive architecture

ACT-R (Anderson, 2007) is a computational representation of a theory of mind[1]. It represents a set of assumptions about the nature of memory and of cognition, and the relationship between them, including time and capacity constraints on computation. ACT-R is programmable, allowing models to be built that share the core assumptions of the theory and test hypotheses about how a cognitive task is undertaken.

4.1.1 The structure of ACT-R

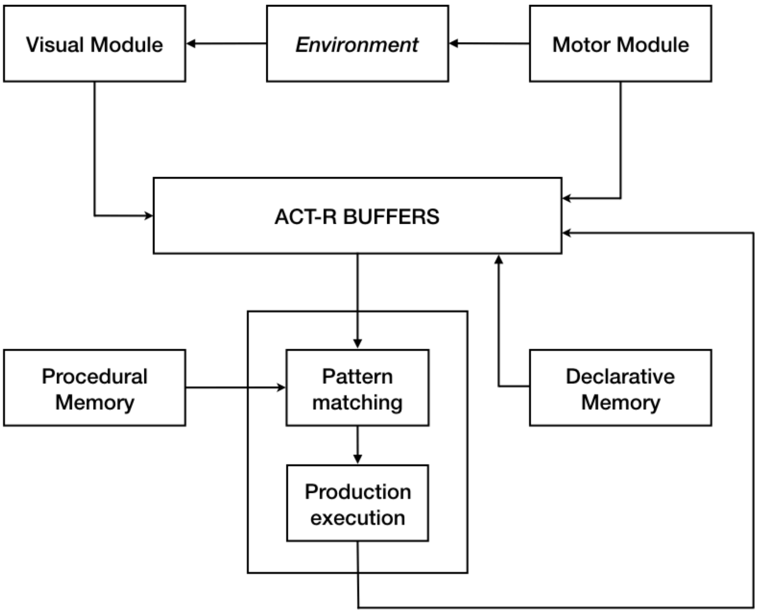
ACT-R has a modular architecture, represented in Figure 4.1. This modularity delineates specific cognitive functions (e.g. visual perception) and allows the relationship between them to be described more exactly. It also supports the generation of hypotheses that localise the functions carried out by a particular module to a specific brain area, which can then be tested empirically. The modular architecture also allows ACT-R to be extended by modellers: the core software covers only a subset of the cognitive tasks that have been studied empirically but template modules are provided for use if these are theoretically or empirically motivated. Many of the ACT-R parsing models make use of additional modules: the scope and use of these is discussed further in Section 4.2.

4.1.1.1 The nature of modules

There are two classes of module in core ACT-R: memory modules and perceptual-motor modules. Memory modules correspond to internal cognitive tasks within the brain. Perceptual-motor modules correspond to the interface between ACT-R and the real world, and include representations of visual or auditory sensation, and of the controls required to move part of the body. Each module has at least one buffer which can hold one piece of information.

---

[1] Technical information in this section is taken from Bothell (2017).



Reproduced from `http://act-r.psy.cmu.edu/about/`

Figure 4.1: ACT-R modular architecture

Although many pieces of information may be generated by or available to a module, only the information that is held in the buffers at a particular point can be processed. This corresponds to empirically observed constraints on the scope of attention and on working memory. To perform complex computations, ACT-R must move information into and out of buffers: the modeller's task is to design structures for information units and a cognitive processing pathway that fits with experimental evidence.

4.1.1.2 Representing knowledge

The memory modules of ACT-R represent knowledge as two distinct types of memory, declarative and procedural. Declarative memory holds knowledge about facts or concepts. Procedural memory governs the ability of ACT-R to manipulate and create declarative memory chunks, and to carry out motor actions. Both types of memory are formed of discrete units, known as *chunks* in declarative memory and *productions* in procedural memory. The discrete nature of the memory chunks themselves and their contents can be described as *symbolic* cognition, but individual chunks and productions also have *sub-symbolic* attributes with continuous values. These sub-symbolic properties can affect the retrieval of declarative memory chunks and the choice of procedural memory productions: parameters relating to sub-symbolic properties can be set by the modeller for the model as a whole, or when chunks are created, but cannot otherwise be manipulated by the modeller during a model's run. ACT-R can thus be described as a hybrid architecture, rather than purely symbolic.

Declarative memory is created and accessed using the core Imaginal and Retrieval buffers. The Imaginal buffer creates new memory chunks that hold a specified set of information, and the Retrieval buffer is used to search for previously-created declarative memory chunks, holding the chunk that most closely matches a specified set of search cues. Procedural memory productions are specified at the start of the model, and can be created during the operation of a model by compiling sequences of productions that fire in sequence. There is also a Goal buffer which holds information to support the processing task or to guide a particular sequence of processing sub-tasks.

Declarative memory chunks are sets of attributes (termed *slots*) with associated *values*. The value of a slot may be a memory chunk, a string, a number, or logical True/False. ACT-R itself does not limit the number of slots within a chunk. However, the underlying theory is grounded in an assumption that memory chunks represent small discrete elements of knowledge such as *Paris is the capital of France* or $3 + 2 = 5$, and accordingly a parsimonious hypothesis should aim to have the smallest possible chunk size which enables the model to fit to empirical data. Slots are also unordered, so if a model assumes an ordered relationship between a number of slots in a chunk, this has an impact on the number of productions required for effective computation. For example, in a chunk that has slots named SLOT1, SLOT2, SLOT3, the names of the slots are arbitrary tokens, even though they contain numbers which can be recognised as a sequence. Accordingly, it is not possible to write a single production that would specify SLOTX, SLOTY, X>Y in order to select any two of those slots with one condition relating to the higher-ordered slot and the other relating to the lower-ordered slot. Instead, separate productions must be written to allow pairwise comparison of the slots SLOT2 vs. SLOT1, SLOT3 vs. SLOT1, SLOT3 vs. SLOT2.

Productions in procedural memory are essentially if-then statements. A production has two parts: a set of conditions which apply to specified buffers and the chunks contained in them, known as the left-hand side of the production (LHS), and a set of consequent actions, framed as instructions or requests to specified buffers, known as the right-hand side of the production (RHS). The process of choosing and firing productions is discussed in more detail in Section 4.1.2.1.

ACT-R's perceptual-motor modules also use chunks to represent a perceived item or a motion to be carried out. However, these chunks may only be accessed within their own module: processing information relating to these perceptual-motor chunks requires a declarative memory chunk to be created in response to the presence of a perceptual-motor chunk in a buffer.

4.1.2 The operation of an ACT-R model

For a simulation to take place, a model must be loaded into ACT-R. At this point, the production set and the set of declarative memory chunks is constructed and made available.

A simulation is known as a *run*: at the start of a run, all buffers are cleared and then a chunk is placed into the Goal buffer, representing the desired outcome of the task. Before this point, all buffers are empty and so no production can be selected.

The core of an ACT-R model is the production cycle. A production is selected according to the conditions in the buffers, and 50ms later the production fires. As soon as the production has fired, the conditions in the buffers are assessed again and if a production can be selected, it is. However, if there is a delay between a production firing and its consequence affecting the contents of the buffers (e.g. time taken for a chunk to be placed in the Retrieval buffer, time to create a new chunk in the Imaginal buffer), the conditions for another production to fire may not be met until the action is complete. Where there is a delay due to a time consequence, ACT-R records the relevant buffer's status as "busy": modellers may use buffer status queries to inhibit other productions firing while a chunk is being retrieved or created. After all of the consequences of a production are complete, if no production matches the buffer conditions, the model terminates.

4.1.2.1 Choosing productions

Productions are selected by the procedural module, which is a pattern-matching system. Each production has two elements: a set of conditions known as the left-hand side (LHS) and a set of consequences known as the right-hand side (RHS). The set of conditions applies to the buffers at the time that a production is chosen. Conditions may be placed on the status of a buffer (e.g. whether or not it contains a chunk, whether or not it is in the process of retrieving a chunk, etc.), or on the contents of a buffer (e.g. whether or not particular slot-value combinations are present in a chunk, or whether a slot value is within a certain range). If the LHS of a production within the production set matches the current buffer status, that production can be selected to fire. The default matching mechanism seeks a full match of all the LHS conditions. However, it is possible to set an ACT-R parameter so that partially-matching productions are available for selection. If no production is available to be selected, the model terminates. If more than one production may be selected, ACT-R makes a choice as to which production fires. If partial matching is available, the production with the highest level of matching is chosen. If two or more productions have the same level of matching, the choice may be random, arbitrarily deterministic (i.e. every time this choice occurs, choose A over B), or dependent on the sub-symbolic *utility* property of the productions. The default state of ACT-R is not to use production utility in pattern matching. If production utility is enabled, each production is assigned a numeric utility value when the model is loaded. If the model includes rewards that are achieved when a particular production fires, members of the production chain that preceded the rewarded production receive additional utility. Increased utility increases the likelihood of the production being selected at a future choice point.

4.1.2.2 Manipulating memory chunks

Chunks can arrive in a buffer in three ways: creation, perception, and retrieval. A request to the Goal or Imaginal buffer causes a new chunk to be created, reflecting the parameters of the request. The time taken to create a new chunk is set by a parameter. For the Imaginal buffer, the default chunk creation time is 200ms, whereas for the Goal buffer, chunk creation is instantaneous by default. Focusing attention on an object using the Visual or Auditory module – the ACT-R equivalent of perception – results in a chunk representing an object in focus being placed in the relevant buffer. Retrieval takes place if a request is made to the Retrieval buffer: a successful request results in a copy of a memory chunk being placed in the Retrieval buffer.

The mechanism for retrieval requests makes use of symbolic pattern matching, as well as the level of activation of individual chunks, which is a sub-symbolic property of declarative memory. Pattern matching takes place according to the search cues contained in the request. A search cue is a set of slot-value pairs, which is compared with all chunks in declarative memory. If a memory chunk has slots and values that match the search cue, it may be retrieved, but this is dependent on the chunk's level of activation.

Chunks gain activation in three ways. When a chunk is created, either when the model is initialised or during a run, it is assigned a baseline level of activation. Retrieval of a chunk results in its activation level being increased. Chunks may also gain what is termed *spreading activation*, if this is enabled by a modeller. For a chunk in a buffer, spreading activation is added to any memory chunk that is the value of one of its slots. This is termed the *fan effect*: the size of the fan being the number of competing memory chunks that receive spreading activation. The overall amount of spreading activation is fixed by the modeller and is shared between those memory chunks that are eligible to receive it. Thus, the larger the fan, the smaller the amount of spreading activation that each chunk receives. One way to model individual differences in working memory capacity within ACT-R is to vary the overall level of spreading activation.

One parameter of ACT-R is the activation threshold, which determines the minimum level of activation required for a retrieval to take place. If there is no chunk that matches the search cues, or if a chunk matches the cues but has an activation level below the threshold, the retrieval fails. Where more than one chunk matches the search cue and is sufficiently activated, the chunk with the highest level of activation is retrieved. Besides governing retrieval choice, the level of activation also determines the time that it takes for a retrieved chunk to be placed into the buffer: the higher the level of activation, the faster the retrieval. Thus the activation threshold can also be seen as corresponding to a "time-out" limit on retrievals.

4.1.2.3 Learning

ACT-R is able to model the process of learning during tasks. Learning takes three forms, two relating to procedural memory, and one relating to declarative memory. The first form of learning, utility learning, was discussed in Section 4.1.2.1. Utility learning increases the probability that a previously-successful production will be chosen, and may therefore result in a more efficient processing pathway being chosen. However, in itself it does not change the time-course of the production that is fired.

Procedural learning can also take place using the *production compilation* mechanism. When production compilation is enabled, a production that fires is considered alongside the production that fired previously. If the starting conditions of the two productions are congruent and there are no blocking circumstances relating to the LHS tests or RHS actions in either production, ACT-R creates a new production that combines the two, allowing the two to fire within one 50ms production cycle. This new production is available for selection and competes against its parent productions, gaining utility either when it is selected, or when the two parent productions fire again in sequence. Productions created using the compilation mechanism may themselves be compiled. Thus cognitive tasks that initially require many productions may subsequently be carried out much faster. The impact of production complication learning on speed of cognition is experienced in 50ms incremental reductions in time-course.

Learning in declarative memory is reflected in increased activation for frequently-used chunks: if a chunk is retrieved often enough, its level of activation may be such that is is always available quickly.

4.1.2.4 After a simulation has terminated

Once a model has terminated because no further productions can be fired, the contents of declarative and procedural memory, including the levels of activation and utility and any compiled productions or created chunks, are still available to be queried. The modeller may choose to maintain declarative and procedural memory for the subsequent run even though the buffers are cleared. This allows learning to take place over multiple simulations. Alternatively, it is possible to reset memory to its initial state.

4.1.3 Model outputs

ACT-R allows experimenters to model various metrics relating to a cognitive activity. Time-course data is available for: production firing; chunk activation levels; and buffer activity. Thus if a model is simulating an experiment with a time measure, e.g. a self-paced reading task, the time interval between firings of the production that moves attention from one word to the next can be taken as a proxy for the experimental time-course, an assumption which is made by several of the models discussed in Section 4.2. Chunk activation levels can also be plotted over time, to examine how retrieval processes and spreading activation influence working memory. The ACT-R theory proposes a link between buffers in the core model and areas of the brain. This hypothesis can be tested by comparing time-course measurements of buffer activity with measures of brain activity such as MEG, EEG, or fMRI.

ACT-R also allows modellers to examine the outcome of a computation. For example, in a task to model the addition of two numbers, the modeller would test whether the result generated by ACT-R is the same as that calculated by human subjects. For a language parsing task, this might include the final structural representation, or the antecedent that is selected during a pronoun resolution task (e.g. van Rij et al., 2012).

4.2 ACT-R and language processing

From its earliest days, ACT-R has been used model language processing tasks. Often these test whether the ACT-R model of general cognitive processes can be fit to experimental data from specific language processing tasks, but researchers have also used ACT-R's capabilities to perform computational tasks in a cognitively plausible way, without specifically modelling processing time-courses. In Sections 4.2.2–4.2.5 I discuss models of incremental language structure-building in more detail, but before then I will briefly review literature on ACT-R models of other aspects of language processing.

The storage and retrieval of lexical memory was considered by van Rijn and Anderson (2003), who modelled the lexical decision task, determining whether a sequence of four letters represented an English word or not. The model fit to experimental data and suggests that no additional memory structure or retrieval mechanism is required to model the lexical decision task, over and above the assumptions of core ACT-R. Van Maanen et al. (2009) considered the interface between lexical access and visual processing, modelling picture-word interference and the Stroop effect. Subsequently Dutta and Basu (2012) used ACT-R as the basis for a word-sense disambiguation network, without testing a particular cognitive hypothesis.

Authors have tested ACT-R's capability with regard to Optimality Theory (Prince and Smolensky, 2002). Misker and Anderson (2003) showed that the learning of Optimality Theory (OT) constraints can be modelled in respect of past-tense formation, and in respect of syllabification. Subsequently Hendriks et al. (2007) applied OT constraints to model the acquisition of constraints on pronoun referents.

Resolution of anaphora, whether pronominal or nominal, has been modelled by authors including Emond (1999) and Pyke et al. (2007). Van Rij (2012) describes four ACT-R models considering aspects of anaphora resolution, including the acquisition of syntactic constraints on referents and the role of OT. Her models cover: production-comprehension asymmetries during the acquisition of reflexive pronoun reference constraints; the role of discourse context in anaphor resolution; differences between the preferences of adults and children after a topic switch during discourse (van Rij et al., 2011); and the relative salience of grammatical and discourse constraints in the resolution of ambiguous pronouns.

The ACT-R models built by van Rij do not create a full grammatical representation of the sentence: standard sentence structures are used, allowing grammatical constraints to be treated as part of the linear order of the input. ACT-R parameters such as spreading activation and the production architecture are used to simulate differences in working memory capacity, grammatical knowledge, and Optimality Theory constraints.

Van Rij uses the model fit with empirical data as evidence to support claims that syntax alone does not constrain anaphor resolution, and that the discourse context can constrain candidates and support early resolution. She also demonstrates that a single set of constraints can produce different processing outcomes depending on the directionality of the processing task (production vs. comprehension).

4.2.1 ACT-R's suitability for models of parsing

The case for ACT-R as a suitable environment for models of parsing was made by Lewis (1999). ACT-R represents a general theory of cognition and memory, using a set of architectural characteristics and constraints. Lewis argues that this architecture interacts with the domain-specific constraints imposed by language structure, and that the results of these interactions account for a range of observed phenomena relating to structural preferences and time-course variations during parsing, and that models that make this assumption have been successful in replicating empirical data, as will be discussed in more detail later in this chapter.

Lewis postulates that cue-based retrieval is the fundamental mechanism used during the parsing process, and that this retrieval is from a memory store whose contents have differing levels of activation. This hypothesis is the theoretical basis for a unified account of empirical data including frequency and recency effects, the impact of syntactic complexity, and time-course and structural preference effects seen in situations of lexical and/or syntactic ambiguity. An implementation in ACT-R explicitly links activation levels to retrieval times: the model is presented as following proposals of race-based parsing and Minimal Attachment put forward by Frazier and Fodor (1978).

The level of activation is influenced by three principles which are general properties of memory: the focus of attention, activation decay, and interference in retrieval. The principle of attention focus requires that memories that are currently the focus of processing activity are highly activated. The principle of decay is that, if a memory leaves the focus of attention, its level of activation decays over time. The principle of interference assumes that retrieval cues provide additional activation to the relevant candidates that are related to memories in focus. Where there are many candidate memories that match retrieval cues, the additional activation is shared between the candidates resulting in lower overall activation for each candidate and hence longer retrieval times.

The ACT-R architecture provides limited focus of attention through a restricted number of buffers and automatic computation of interference and decay effects through its declarative memory module. The task for the modeller is then to design a parsing process of retrieval operations (remembering that each retrieval operation will affect activation levels), representations of linguistic structure that cover the aspects of content that are relevant to the task being modelled, and appropriate search cues which match the requirements of the grammar. At the same time, the modeller should account for proposed increases in ACT-R's buffer capacity, which loosen the constraint on the scope of attentional focus, and affect the distribution of additional activation related to search cues and retrieval processes.

The details of the grammatical assumptions and search cues are not given in Lewis (1999), although they form the basis of subsequently-published work discussed in Section 4.2.2. However, modelling results are presented that show a fit with experimental evidence discussed in Chapter 2 on: subject/object relative clause processing asymmetries and various structural ambiguities including between relative clause/subordinate clause, e.g. (Gibson, 1998); attachment within or to the head of a relative clause, e.g. (Gibson et al., 1996), and other attachment ambiguities, e.g. (Traxler et al., 1998).

The focus of this thesis is on the grammatical assumptions embedded in a model. In the review that follows, I will draw out the information in the modellers' assumed grammar, parsing mechanism including search cues, and any additions or amendments to the architecture.

4.2.2 Models that build structure to test a hypothesis about different aspect of language processing

I begin by discussing some of the earlier models of language processing that test hypotheses about meaning representations and about the relationship between sentence structure and inference. Structural parsing is an integral part of these models, but not the main focus of the research projects.

4.2.2.1 Exploring sentence memory

Anderson et al. (2001) test the hypothesis that it is possible to model phenomena relating to sentence processing without assuming distinct sentence memory storage or retrieval mechanisms. They assume three different levels of representation of sentence information: a structural representation ("surface code"), a propositional representation ("textbase") and a situation model, which includes inferential as well as computed information.

The authors discuss a single model, although the code available[2] contains six distinct model files which have slight architectural differences: each model relates to a distinct set

---

[2] `http://act-r.psy.cmu.edu/wordpress/wp-content/uploads/2013/09/models2.zip`. Last accessed 2019-03-02.

of published experimental data. In each case, the model takes as its input a sequence of sentences which form a short story. For each sentence, the model builds a syntactic and propositional representation, and constructs a relationship between each referent in the propositional representation and a discourse representation that covers the whole story. Outputs of the model are time course data for the processing of words or determiner phrases (e.g. *the-waiter*), and an accuracy assessment of the generated semantic structures against human performance.

On the basis of the model, the authors conclude that memory for processed language content can be encoded and retrieved using the same mechanisms as other types of memory, and that there is no need to postulate a distinct memory retrieval function. They further argue that the level of abstraction of the representations, and the memory-retrieval processes used in building them, can account for observed differences in longer-term recall of sentence structure and content.

Outcomes The authors were able to process structures and generate propositional outcomes at a speed that matches human performance.

Grammatical assumptions The scope of the grammar is a highly restricted subset of English, covering simple transitive sentences in active and passive forms, with no significant structural ambiguities and no embedding. In some of the models, the sentence patterns are extended with adjunct prepositional phrases sentence initially and, optionally, sentence finally. There are thus a total of six sentence patterns (active/passive × 0/1/2 adjuncts) that can be processed, with a stipulated relationship between word order and thematic role.

The authors assume a modified version of X-bar theory (Jackendoff, 1977). Although they do not state the phrase-structure rules they assume, the rules for a transitive sentence in active and passive forms, containing only the verb and NP/PP to represent the agent and patient, can be derived from structural representations provided, and are shown in (177).

(177) a. $\mathrm{S} \rightarrow \mathrm{NP}_{arg}\ \mathrm{VP}_{head}$
b. $\mathrm{VP} \rightarrow \mathrm{V'}_{head}\ \{\mathrm{NP}|\mathrm{PP}\}_{arg}$
c. $\mathrm{V'} \rightarrow (\mathrm{Aux}_{arg})\ \mathrm{V}_{head}$
d. $\mathrm{PP} \rightarrow \mathrm{P}_{arg}\ \mathrm{NP}_{head}$
e. $\mathrm{NP} \rightarrow \mathrm{N}_{head}$

These deviate significantly from the standard assumptions of X′ theory (e.g. the NP object of or PP adjunct to the VP both attaching at VP). It is not clear how robust the assumptions would be if the scope of the grammar were to be extended, whether to include a greater subset of English, or to be adapted for languages with a lower level of configurationality and/or a less constrained relationship between word order and argument structure than the model assumes.

Parsing assumptions The parsing process builds on Lewis (1999). The parsing cycle involves three steps: word recognition; retrieval of a lexical entry; and growth in syntactic and semantic structures. Structure growth involves chunk creation and association of semantic chunks with discourse referents. Parsing is driven by context: intermediate parsing goals are stored in the Goal buffer, and the action to take in response to the next word is dependent on the combination of word category and context. A production in response to one word sets the goal parameters for the next word.

Architectural consequences The models assume that separate buffers exist to construct chunks of each phrasal category (NP,PP,VP,V′,S) and to hold the syntactic and semantic chunks that are produced. Each of the six models includes either 25 or 26 extra buffers beyond the ACT-R core.

Discussion The notion of distinct types of representation (structural, propositional, situation) is intrinsic to the model's success in replicating experimental results. The model also assumes that the different representations are built at different rates according to the availability of information from the sentence. These assumptions on representation are congruent with LFG assumptions on structural representation, which were discussed in Chapter 3.

However, the relationship between structural and propositional representations, both argument structure mapping and the point at which the structure is extended, is stipulated in the production set. If the grammar is seen as a finite state automaton, a single production governs each possible arc, and this production also determines the extent to which propositional content is updated when sentence structure is built. It is unclear how this could be scaled to a grammar that included embedding or recursion, to a wider range of valencies and valency alternations, or to a situation with fewer ordering constraints.

Phrase-specific buffers are used to generate phrase-specific structural chunks. This avoids the need for memory retrievals other than the initial lexical information. However, it results in a proliferation of language-specific buffers beyond ACT-R's core cognitive capacity, and embeds a set of situation-specific phrase-structure assumptions into the architecture of the model.

Overall, the parsing model was fit for the authors' purposes, although it does not fit in detail with any established syntactic theory. Particular points to take into account in attempting an LFG model include support for the notion of multiple levels of representation, and awareness that modelling assumptions on the role and nature of the production set, and of additional buffers, may result in limiting the general applicability of a model.

4.2.2.2 Syntactic and semantic processing

Budiu and Anderson (2004) investigate the relationship between syntactic and semantic processing, in particular the processing of metaphors, and of sentences with so-called "semantic illusions", e.g. the strong tendency of readers to accept the sentence *When an aircraft crashes, where should the survivors be buried?* as semantically unproblematic (Barton and Sanford, 1993). The model is an implementation of the authors' interpretation-based processing theory (INP), which postulates that all language processing, whether literal or metaphorical, uses a single mechanism, and that the processing context, including information provided from the sentence as it unfolds, increases the probability that hearers will construct the intended meaning, even if a metaphor is unknown or the speaker makes a lexical error.

The goal of the model is to match the sentence content to a similar piece of knowledge as quickly as possible. It is explicitly assumed that this match may take place before the sentence is completely processed, in line with the assumptions of Good Enough Processing (Ferreira et al., 2002). Similarly to Anderson et al. (2001), the model assumes multiple levels of representation, with separate syntactic, propositional, and situational representations.

There is little detail given in the paper about the syntactic parsing process, which is in line with the model's focus on testing hypotheses about semantics. However, the code for the models has been made available for analysis[3].

Outcomes The authors acknowledge the difficulties with fitting quantitative experimental outcomes, and instead present evidence that the model outcomes qualitatively match experimental data on the relationship between position of metaphor in a sentence and processing speed, that the model is similar to human subjects in generating "illusory" situational representations of sentences such as *How many animals of each kind did Moses take on the ark?* (Erickson and Mattson, 1981), and that the model replicates "text-priming" effects, where the processing of words semantically similar to previous words in a story is faster than semantically unrelated words.

Grammatical assumptions The scope of the grammar is again restricted, though not as tightly as in Anderson et al. (2001). Grammatical assumptions are stated as being similar to those in that paper: constituent structure chunks represent a binary branching tree with each node/chunk having a HEAD and an ARG daughter. The syntactic structures presented (Budiu and Anderson, 2004, Fig. 2, p. 7) show the use of N' levels for the attachment of adjectives and modifying nouns within noun phrases, but the arguments of VP and PP attach at the XP level rather than at X'.

Parsing assumptions During the parsing process, the Goal buffer tracks the next expected node to be attached. Where the category of the word read meets that expectation, intermediate nodes are generated as necessary, together with any required XP argument nodes. The process appears to be similar to a finite state automaton.

---

[3] `http://act-r.psy.cmu.edu/wordpress/wp-content/uploads/2013/09/ModelFiles.zip`. Last accessed 2019-03-04.

The authors note two processes for managing syntactic ambiguity. One is that there is a delay in assigning a propositional role to a referent in a particular syntactic position until the verb form has been assessed as active or passive, allowing an accurate assignment of agent or patient to the first argument. The second is that syntactic structural relationships between parent and child nodes may be changed during parsing. The authors give an example during the parsing of *the college students were taught by professors* where *college* is initially parsed as the HEAD daughter of node N′1, which is itself the HEAD daughter of node NP1. However, the next word is *students* which becomes the HEAD daughter of node N′1, replacing *college*. The word *college* becomes the HEAD daughter of a node N′2 which is the ARG daughter of node N′1.

Architectural consequences Similarly to Anderson et al. (2001), separate buffers are created for each type of syntactic node, and for particular elements of the semantic and situational representations. From the code it appears that 49 additional buffers are used over and above the core ACT-R buffers, of which 28 relate to chunks used in the syntactic representation. Of the 72 productions in the model, 35 are specific to building the syntactic representation with consequent updating of the propositional representation: these productions each consider a particular combination of word category and buffer context.

Discussion Again, the parsing model is fit for the authors' purposes, although it still does not fit closely with an established syntactic theory. Although the authors claim that their syntactic model could be extended, the rigid positional approach to thematic role assignment would be challenged by, for example the different thematic structures that are found when verbs with a dative alternation are passivised, as in (178).

(178) a. The book$_{theme}$ was given to David$_{goal}$ by Stephen$_{agent}$
b. The book$_{theme}$ was given by Stephen$_{agent}$ to David$_{goal}$
c. David$_{goal}$ was given the book$_{theme}$ by Stephen$_{agent}$

Of more interest is the higher-level process interaction between context, information structure within a parsed sentence, and inferential processing. The INP theory, together with the model's explicit assumption that the desired cognitive parsing outcome is a meaning, rather than a structure, produces a model prediction that inferences will be faster and more accurate if given information precedes new information in a sentence. The model also predicts that activation of a referent during processing will activate related facts, even if no inference was computed at the time of sentence processing. This is helpful in creating a relationship between LFG and processing tasks which will be discussed further in Chapter 5.

4.2.3 Models aiming to extract meaning within time constraints, without testing hypotheses

A number of papers in the literature (Ball, 2003, 2004, 2005, 2007a,b, 2010, 2011a,b, 2012b,a, 2013; Ball et al., 2007; Freiman and Ball, 2010; Heiberg et al., 2007) report on a project whose stated aim is to match human reading and response speeds in words per minute, initially in real computation time and ideally also in model time. Within this overarching aim, however, there is no intent to match more fine-grained experimental data on processing time-courses.

The papers were generated from the U.S.A.F.'s *Synthetic Teammate* project, which was an attempt to model the behaviour of a human aeroplane pilot for use in training. By the final paper, their model had almost 58,000 declarative memory items and was able to process written text at close to 200 words per minute in real time. Publications ceased in 2013 and attempts to contact the authors using their published e-mail addresses failed. As at October 2017 there was no reference to the Mesa Research Centre on the U.S.A.F. website.

Outcomes The project outcome measures are related to accuracy and overall parsing speed, rather than time-course for specific processing tasks. Reporting of actual outcomes is limited: Ball et al. (2007) describes a set of constructions that the model is capable of processing, but no more detail is given.

Grammatical assumptions The papers that refer to grammar use the *Double-R Grammar* theory first described in Ball (2003), which was developed within the project. The grammar builds on elements of X-bar theory (Jackendoff, 1977), Simpler Syntax (Culicover and Jackendoff, 2005), Sign-Based Construction Grammar (Boas and Sag, 2012), and Cognitive Grammar (Langacker, 1986, 1987, 1991) but also includes insights from the project team motivated by the technical challenges of parsing text using ACT-R. I have found no citations for the theory by authors outside the project. The theory is described only for English and leans heavily on English word-ordering constraints in building syntactic structure and assigning thematic role.

Parsing assumptions Ball (2011b) gives the most recent overview of the parsing model and describes a parsing cycle with the stages activation, selection, and integration. A fourth stage, *context accommodation*, allows for the rearrangement of recently-integrated structure where subsequent context shows that the initial analysis is no longer possible. However, the model cannot repair sentences that produce garden-path effects in humans.

Architectural consequences Ball (2011a) and Ball (2012a) both discuss issues related to memory structure at the level of the individual chunk, and also at the level of working memory capacity. At the individual chunk level, they have identified the difficulties inherent in ACT-R of modelling the various dimensions (phonological, semantic, syntactic) in which a processed word spreads activation to related words. Their solution required additional Lisp coding within ACT-R productions, because the ACT-R chunk type hierarchy supports only spreading activation in one dimension.

With regard to working memory capacity overall, the authors propose numerous other buffers to create and store particular types of chunk. The motivation for these buffers is taken from the model's functioning rather than specific evidence from neuroscience. Some of the capacity, such as specific buffers to create chunks of each word class, is similar to the approach of Lewis and Vasishth (2005). Specific buffers are also proposed for grammatical functions (Ball, 2013). However, the authors also claim, *contra* Lewis and Vasishth, that specific buffers are required to hold partially-processed structures, because otherwise retrieval time costs are too high[4]. For example, their buffer WH-FOCUS is required only in languages with wh-fronting, with the suggestion that speakers of wh-in-situ languages do not need to develop this capacity.

Discussion The papers are interesting because of the detailed discussion of issues relating to the architecture. The authors assume a maximally incremental and interactive parsing process with the high-level time constraint of matching human speech, and so the challenges they describe are likely to be encountered in any project to cognitively model language processing. However, the limitations of Double R Grammar, including its specificity to English, hinder comparisons between it and LFG. The two models in Section 4.2.2 proposed buffers based on an X-bar grammar of English. This project commits its architecture even more specifically to English, with the authors acknowledging that buffers such as WH-FOCUS may not be valid or required in other languages. If ACT-R buffers are taken as reflecting human functioning to some extent, this choice makes implicit claims about the impact of language acquisition on cognitive structures and processes, which is a position that I am seeking to avoid in developing a universal LFG-based model.

4.2.4 Lewis and Vasishth (2005): Testing hypotheses about the nature of structure-building

The most influential ACT-R model of structure-building is Lewis and Vasishth (2005) (henceforth L&V05), described by Parker et al. (2017, p. 134) as "currently the only mathematically precise expression of a content-based retrieval theory to be applied to psycholinguistic data to date." The model tests a hypothesis on the nature of cue-based retrieval in developing complex syntactic structures including sentences with non-local dependencies and garden-path phenomena. As ACT-R includes a parameter for activation 'noise', it is possible to carry out repeated simulations of ambiguous constructions to model

---

[4] The authors also give a caveat that the buffers in their model that hold partially-processed structures may be language-specific.

gradient phenomena. The contribution of other elements of processing such as semantics is not included, but the authors claim that the gaps between their modelled results and empirical data are likely to be reduced as other elements of processing activation are included.

The model outputs a structural representation of an input sentence, and generates a time-course allowing the processing time of individual words to be modelled and matched against experimental data. Because the model's purpose is to model how the interference effects arising from sentence context have an impact on cue-based retrieval times, the specifics of the grammar are secondary to the need to have the appropriate retrieval cues used at appropriate points during the sentence. Using English-specific phrase-structure rules contained in procedural memory, the model was successful in replicating empirical time courses for a range of phenomena in English, including Subject RC vs Object RC asymmetries (Grodner and Gibson, 2005), working memory retrieval time estimates (McElree et al., 2003), the processing impact of interpolated materials on main verbs and embedded verbs (Grodner and Gibson, 2005), effects of length and interference in garden path sentences (compared with unambiguous control equivalents), and the effects of storage load of complex syntax (Grodner et al., 2002). The authors also model a taxonomy of the relative processing difficulty of various types of centre-embedding in English.

Based on the fit between the model outputs and empirical time-courses, the authors claim support for the use of cue-based retrieval, rather than serial retrieval, in syntactic processing. This includes cues that are based on the calculation of syntactic structure, specifically the c-command relationship (Reinhart, 1976). The model also supports the claim that there is a single cue-based retrieval mechanism that operates in all circumstances, rather than requiring the assumption of different mechanisms and different sets in different circumstances.

4.2.4.1 Extensions of L&V05

The success of the model in English allowed the construction of computational models outside ACT-R (e.g. Engelmann, 2015) that can generate ACT-R internal metrics (e.g. chunk activation and hence retrieval time and outcome) based on the model's cue-based retrieval assumptions and parsing algorithm, but permitting different assumptions on the specific phrase-structure rules. In other words, the mathematical models can generate time-courses for supposed ACT-R retrievals, based on the assumption that structure is being created according to a certain parsing method, but without the need to write a new set of ACT-R productions or to generate a structure by running a model in ACT-R. This has considerable benefits in reducing the time required to generate model measures and has been used by a number of researchers testing and refining L&V05's claims on cue-based retrieval. The original structure-generating model has also been expanded with the development of an eye-tracking module (Engelmann, 2016) that outputs model time-courses for eye movements during eye-tracking experiments. This allows finer-grained matching of the overall model time course against empirical data.

The extensions of L&V05 have allowed testing of hypotheses about cue-based retrieval for complex syntax in other languages, and phenomena where empirical results suggest that there is interference between retrieval cues, in English and other languages. Interference occurs in cases where competitor chunks for attachment (e.g. during subject-verb agreement) or competitor referents (e.g. antecedents of reflexive anaphors) partially match the structural and/or feature sets of the intended attachment site or referent, and the processing time-course varies according to the particular combination of match and mismatch.

Jäger et al. (2015) consider reflexive anaphora in Mandarin and find that neither structural calculations nor L&V05 as presented can account for their experimental data. They propose an enhanced model based on L&V05 but with two additional factors that can modulate the cue-based response. One of these, prominence, relates to characteristics of the competitors and aims to capture activation differences which might be based on functional syntactic factors such as the relative position of competitors on the grammatical function hierarchy (Keenan and Comrie, 1977), discourse and information structure factors such as topic or focus, and semantic features such as animacy or thematic role. The other factor, cue confusion, assumes that the features used in search cues may not be discrete, but instead may be continuous variables with multiple input sources. This leads to predictions that specific search cues may have different interference profiles, with more specific cues having a greater power to interfere: the authors contrast interference profiles between structurally and semantically constrained *herself* or *themselves* in English and the less-constrained Mandarin *ziji*.

Dillon et al. (2013) compare syntactic structural cues alongside semantic/syntactic feature cues for the processing of subject-verb agreement and for seeking the antecedent of reflexive anaphors. Experimental evidence was gathered comparing the impact of number feature mismatch between subject and verb, and between anaphor and antecedent, on processing time-courses. Dillon found that the impact of mismatch showed a different profile between the two types of dependency (described by Dillon as "grammatical function"): for reflexive anaphors, intrusion was seen only when the mismatched candidate satisfied the syntactic structural constraint on antecedents, whereas for subject-verb agreement, syntactic structural constraints did not affect the impact of the mismatch. This finding is given as evidence that retrieval cues operate differently between the two phenomena.

Kush and Phillips (2014) also consider the processing of reflexive anaphors, exploring the impact of the relative position of the verb and the anaphor on the processing time-course by collecting data on Hindi reflexives, where the verb is sentence-final, and comparing this with the productions of a computational model based on L&V05. In their experiment, distractors were positioned between the antecedent and the reflexive, within an embedded relative clause modifying a locative adjunct noun. Empirical data were not in line with the model's prediction. The authors conclude that this is most likely to be because structural cues are prioritised over semantic featural cues during retrieval, and because the sentence structure leads to a garden path phenomenon requiring reanalysis of the distractor from the matrix clause into the embedded clause. However, they note also the possibility that cues may be confusable as proposed by Jäger et al. (2015).

Parker (2014) uses the mathematical basis of L&V05, together with weighting of structural cues for computational modelling of retrieval interference in phenomena described as "grammatical illusions," considering interactions between structural and feature-based retrieval cues. He concludes that, while both structural constraints and retrieval cues are used in memory retrieval, structural information takes priority. Furthermore, the nature of the task for which retrievals are taking place, e.g. whether structure is being built, or whether a retrieval is for reanalysis or error correction, also has an impact on interference.

Patil et al. (2016) also consider processing of reflexive anaphora and make use of a model based on L&V05 to predict the effect of both structural and non-structural cues being used together in retrievals. These predictions are supported by the authors' empirical data: the authors suggest that the lack of reported effects from non-structural cues in previous studies may be the result of a smaller effect size for non-structural cues and corresponding lack of statistical power.

Jäger et al. (2017) compare the predictions from L&V05 against a meta-analysis of empirical data drawn from investigations of cue-based retrieval in the processing of dependencies. They specifically consider number cues in subject-verb agreement, animacy and gender cues in subject-verb selection dependencies (where these are not involved in syntactic agreement but might influence the choice of an attachment site – a set of cues termed *non-agreement* by the authors), and number and gender cues in antecedent-reflexive agreement. Their findings are shown in Table 4.1, reproduced from Jäger et al. (2017, p.332):

Table 4.1: Jäger et al. (2017) assessment of predictions from L&V05

Dependency	Target	L&V05 prediction	Supported?
S-V (non-agreement)	Match	Inhibition	Yes
S-V agreement	Match	Inhibition	No
	Mismatch	Facilitation	Yes
Reflexive-/reciprocal-antecedent	Match	Inhibition	No
	Mismatch	Facilitation	No

The authors suggest three possible reasons for the lack of fit between the predictions of L&V05 and the empirical data. One possibility is that cue-based retrieval is not involved in the construction of dependencies during sentence processing. However, other proposed theories have not been as successful in fitting the data, and using the mechanism to build sentence structure is in line with more general theories of memory and cognition. A second possibility, echoing Patil et al. (2016), is that lack of statistical power may make the empirical findings unreliable. A third possibility, preferred by the authors, is that the L&V05 model must be revised to allow finer-grained predictions for sentence processing. Specific suggestions are that the different types of dependency require distinct assumptions about their processing; that additional factors beyond syntactic and semantic cues are required to explain the observed inhibitory interference; and that the processing advantage of cue-matching in agreement dependencies must also be explained.

In summary, the development of a mathematically-precise model by Lewis and Vasishth has had a significant impact on theoretical debate, providing support for accounts that place sentence processing within wider theories of memory access and cognition, enabling the elaboration of more detailed theories of the nature of cues and the purposes of retrieval, supporting methodological challenges to earlier work, and allowing the limits of validity of the model itself to be identified.

4.2.4.2 Features of L&V05

As with the other models discussed earlier in this section, I now turn to the role of grammar and the processing architecture that was assumed in the construction of the model.

Outcomes The model generates a representation of syntactic structure in the form of a set of memory chunks, each of which forms a node in a constituent structure tree. The structure is not maintained in working memory, but is recoverable at the end of a parse. Measures of the modelled time-course are also generated, including the total time for a parsing cycle for each word, and the retrieval time for the attachment site for a particular word.

Grammatical assumptions The model assumes a modified version of X-bar theory (Jackendoff, 1977; Chomsky, 1986) where the root node for a sentence is IP. The grammatical representation is a set of binary-branching tree nodes conforming to a set of phrase-structure rules. The specific rules used by the model are not provided in the paper, but can be derived by examining the productions within the ACT-R module included as part of the code published by Engelmann (2016). Lexical items project phrases, and a phrase attaches into the overall structure from its maximal projection.

Parsing assumptions The module uses a left-corner parsing algorithm (Aho and Ullman, 1972). Each word is assumed to be attached into the structure before the subsequent word is attended, with intermediate structure being created as necessary if a direct attachment point is not available. Each node category (combination of word class and bar level) is represented by a distinct ACT-R chunk type. The presence of an unresolved long-distance dependency and other grammatical information that is not immediately contained in the syntactic tree (e.g. c-command status) is available to the parser through information stored in the Goal buffer. At a high level, the parsing cycle is as follows.

(i) Attend a word displayed on the screen and retrieve its lexical entry. Store a representation of this in the Lexical buffer.

(ii) By comparing the retrieved lexical entry and the expected category (see step iv), select and fire a production that uses structural and semantic retrieval cues to find a previously created syntactic chunk where the new word can attach. On the basis of this production, only the attachment site is retrieved. The time taken to retrieve the attachment site varies according to the level of activation of the attachment site chunk and the total number of chunks that are competing for retrieval.

(iii) Once the chunk representing the attachment site is retrieved, attach the chunk in the Lexical buffer, creating any additional intermediate syntactic structure that is required for attachment to take place.

(iv) Update the predictive rule on basis of the sentence processed thus far, which gives an expectation of the category to be retrieved. The expected category is stored in a slot in the Goal buffer. Release the new memory chunk and its attachment site from working memory.

Architectural consequences The authors explicitly address the architectural choices that underpin the design of the model. Procedural memory stores a detailed set of phrase-structure rules as parsing operations. Declarative memory holds lexical specifications and the representation of syntactic structure. The Goal buffer holds information that controls the parsing process, and also stores elements of grammatical information that are not immediately contained in the tree. The c-command relationship between chunks is calculated using an internal Lisp function and does not require additional structural retrievals.

The model assumes significant additional buffer capacity, although much of this is not explicitly available to modellers. Chunks to hold lexical information at leaf nodes of the tree are generated by a Lexical buffer. Although the retrieval of lexical specifications and the retrieval of attachment sites uses the core ACT-R Retrieval buffer, additional Goal-style buffers are assumed that are specific to particular categories of tree node (e.g. buffers named VPb, NPb, DPb, CPb for verb phrases, noun phrases, determiner phrases and complementiser phrases respectively), and these are used to create the intermediate structural nodes between the chunk representing the leaf node and its attachment site.

The presence of an embedded sentence affects the phrase-structure rules that are used to set search criteria: structurally, the nodes within an embedded clause have a different type to nodes within the matrix clause. This is reflected in specific buffers, chunk types, and productions for matrix, embedded, and gapped clauses.

Although the model uses a left-corner parsing algorithm, there is no explicit stack of phrase-structure rules. Instead, the function of a stack is fulfilled by information in the production rules which states what the expected next category is to be processed. This category may not match the next word encountered: the combination of the two determines the next production rule to be applied.

Discussion The model successfully replicates a range of empirical findings using general assumptions about the operation of working memory, adjustment of only one ACT-R memory parameter, and adding a small amount of overt working memory capacity in the form of additional buffers. It thus offers a more parsimonious account of sentence processing than theories that hypothesise discrete and distinct language processing modules. Its cue-based retrieval architecture also provides a robust alternative to structural theories of processing complexity, with the authors identifying improvements compared with the Dependency Locality Theory (Gibson, 1998, 2000).

However, in addition to the overt working memory capacity, the model includes significant hidden capacity to create intermediate structural chunks. The authors share an assumption with some of the models previously discussed, that structural chunks are created by buffers that are specific to a particular category. The set of categories assumed in the grammar – including the distinct set of categories for use in embedded and gapped clauses – is thus reflected in the total amount of hidden working memory capacity.

The assumption of specific phrasal categories for phrases within various clause types (e.g. VP, $\mathrm{VP}_{embedded}$, VP_{gapped}) can be seen as one consequence of dispensing with the stack of phrase structure rules commonly assumed for left-corner parsing. Where a stack is present, it holds the planned route through the parse and allows this to be amended in the light of the unfolding sentence. Without a stack, it must still be possible for embedded clauses to be distinguished from the matrix clause, and parallel categories for the different clause types is one way to achieve this. It should be noted that not all relative clauses have a missing argument (e.g. *the day we went fishing*, where the head noun has an adjunct function within the RC), and that there may be temporary ambiguity between embedded clause types where a noun can select a complement clause, (e.g. noun + RC in *the news that David told us* vs. noun + complement in *the news that David had baked a cake*).

As well as replacing the stack, the specific category set also functions to reduce the chance of misretrieval of an attachment site during processing. By restricting the search cue to eligible chunks within an embedded or gapped clause, any interference from potential attachment sites in the matrix clause is removed.

Linguistically, the assumptions about the categories necessary to parse, and the translation of the phrase-structure grammar into a production set, are specific to English. The assumptions of chunk activation for L&V05 have been successfully applied to mathematical models for other languages, but considerable additional work would be required to implement a structure-building model, particularly in typologically different languages.

Finally, the model focuses solely on the construction of a syntactic representation. The authors discuss the possibility of expanding the model to the construction of semantic or discourse representations similar to the models discussed earlier. This could potentially address those situations where the model time-courses are faster than observed times, but raises the question as to the linking hypotheses between the X-bar theory of syntactic structure assumed by the author and its relation to semantic and discourse structures.

4.2.5 Other structure-building models in ACT-R and other cognitive environments

Besides the ACT-R models of structure-building during perception discussed above, there are other models of language comprehension for which details of the operation are not yet available, or of structure-building during production, or models of comprehension which use other cognitive architectures. These are discussed briefly here.

4.2.5.1 Structure building using a left-corner parser with a restricted stack

Brasoveanu and Dotlačil (2018) report the development of an ACT-R model of a left-corner parser which includes specific interfaces with visual and motor modules. However, at the time of writing the full details of the model are unpublished.

4.2.5.2 Structure-building in speech production

Reitter et al. (2011) specifically consider the role of grammar in language processing, but with the processing task being production rather than comprehension. The model was designed to investigate the cognitive mechanisms underlying syntactic priming.

The model is based on a specific grammatical theory, Combinatory Categorial Grammar (Steedman, 2000) which uses the notion of syntactic types to constrain grammaticality. In this theory words and sentence fragments, whether or not they are complete constituents, have syntactic types. The lexical specification for a word gives information about the other syntactic elements that are required to build a complete sentence (for verbs) or a complete noun phrase (for nouns), together with the ordering constraints on the presentation of those elements in relation to the word. Each new word encountered is processed in the context of the preceding portion of the sentence, allowing the types of both elements to be compared. Adjoining the new word and combining it with the context provides an updated context with a new type.

In the model, no additional buffer capacity is proposed: a chunk in the Goal buffer is used to hold chunks representing syntactic type as well as information about semantic roles in the sentence being generated.

The authors found that if a model allows for learning individual syntactic representations, and allows for links to be made between these representations and individual lexical items or semantic representations, then it can replicate syntactic priming effects without requiring specialised cognitive processes.

4.2.5.3 Models of language processing in cognitive environments other than ACT-R

Hale (2014) demonstrates how a generalized left-corner parsing algorithm can be implemented in the Soar architecture and uses it to replicate the results of garden-path sentences. The Soar architecture (Laird, 2012) implements the Problem-Space Computational Model developed by Newell et al. (1991), which is described as a "general theory of decision making and problem solving". Alongside declarative and procedural memory types, it also contains semantic and episodic memory. However, its basic processing cycle – including the selection and firing of processing operators and the activation and retrieval of memory chunks – appears to be similar to ACT-R. Unlike ACT-R, sequential knowledge is a distinct form of knowledge in the Soar architecture, realised through an explicit episodic memory, and the architecture also includes a goal stack, unlike ACT-R.

Lindes and Laird (2016, 2017), also working within the Soar architecture, introduce the Lucia parser. The parser is based on Embodied Construction Grammar (Bergen and Chang, 2013), a formalism which makes use of schemas that describe structure and semantics, including mappings from syntactic structures to actions and thematic roles. The output of the parser is a representation of meaning that can be comprehended by an embodied cognitive agent (Mohan et al., 2013): intermediate syntactic representations are produced to enable the construction of meaning.

Stewart et al. (2014) implements a left-corner parser in Nengo (Eliasmith, 2013), a sub-symbolic architecture that simulates the firing patterns of neurons in the brain. Within the Nengo architecture, connections are made between groups of neurons which allow the modelling of memory maintenance and decay processes, together with pattern-matching, decision-making and learning. Symbolic concepts (analogous to chunks in ACT-R) are implemented through Nengo's Semantic Pointer Architecture, by which a multidimensional vector representation of a concept maps to a firing pattern of a group of neurons. Stewart's parser implements the parsing algorithm and includes a stack within memory that is subject to memory decay processes. The reported model can parse simple sentences with a degree of accuracy, but does not have the capacity to use context to assess attachment ambiguities or to repair a parse if the wrong rule is chosen.

4.3 Discussion

In this discussion I address three points that arise from the models reviewed in Sections 4.1 and 4.2. These include the nature of the mental representation of structure being developed, the role of grammar in the overall process, and the generalisability of architectural modelling assumptions.

4.3.1 The nature of the representation

The focus on syntactic structure in L&V05 allows for a more detailed treatment of the structure than in earlier, multi-tiered models. However, because the relationship between tiers of representation is important in modelling discourse effects, the model lacks the capacity to model how non-syntactic processing might influence the time-course: this is explicitly recognised in L&V05 where the impact of extending the model is discussed. The exclusive focus on syntactic constituent structure and attachment may mask other processing factors: as Dillon et al. (2013, p. 101) point out: "Another possibility is that reflexive and agreement dependencies are computed at distinct levels of representation in the course of online processing."

A problem for ACT-R is the discrepancy between the rate at which words are perceived in natural language, where words may be read at a pace of 240 words per minute, or one word on average every 250ms, and the empirical timescales observed for processing activity related to a particular word, which may occur 600ms or more after the word is initially perceived. The models reviewed seek to process a single word before the next is attended. In the multi-tiered models of Anderson et al. (2001) and Budiu and Anderson (2004) a highly-stipulative structure-building process delivers a rapid result which allows time for the other tiers of representation to be considered within the constraints of observed processing speed. This is at the cost of the grammatical scope and flexibility of the utterances which the model is able to process. The more flexible structure-building approach of L&V05 takes longer, delivering a model processing speed for a structural representation that fits empirical processing speeds. It may be that the human comprehenders are only able to move to the next word once the input word has been added into a basic syntactic structure, but this then implies that – in the absence of a phonological loop – the later stages of processing word$_n$ occur in parallel with the initial processing of word$_{n+1}$. Modelling separate simultaneous processes is complex, and constraining mental representations so that they can be generated in a single step may fit the purposes of a particular subset of the grammar, but it has an impact on the generalisability of the model, which I now turn to.

4.3.2 The role of grammar

The models reviewed were focusing on modelling psycholinguistic phenomena, and so from the modeller's perspective, the choice of grammar was perhaps less important than the ability to deliver overall parsing times (and variations in parsing speed) that match the requirements of the experimental data. Thus Anderson et al. (2001) and Budiu and Anderson (2004) are interested in building a propositional representation that can be tested, and the syntactic structure is a means to an end. For L&V05, Hale (2014), and Stewart et al. (2014) syntactic structure is at the heart of their research question, but the model does not generate a propositional representation of the sentence input. Provided that there is some theoretical support for the phrase-structure grammar and parsing algorithm, the relationship of this to compositional semantics or to discourse representations is less important.

However, if we are to understand the role of grammar in processing, a more robust approach to grammar in modelling is necessary. We need a grammar that is capable of describing constraints across the range of processing tasks, and which can be applied consistently to a variety of processing phenomena. Syntactic structure-building – whether using phrase-structure rules or construction grammars – is one element of language processing, but it does not describe the totality of processing and nor it empirically certain that a full, up-to-date syntactic representation is generated and maintained once initial structure has been assigned (e.g. Christianson et al., 2001). LFG, as a modular theory that distinguishes between different elements of linguistic analysis, and which is capable of accounting for frequently-observed ungrammatical utterances (e.g. the account of production of resumptive pronouns in English by Asudeh, 2012) offers the opportunity to develop a more comprehensive account of grammar in processing.

4.3.3 The generalisability of modelling assumptions

There is a fundamental question about the generalisability of a particular model. All the models we have seen are modelling experiments whose stimuli are drawn from a subset of one language's grammar, with the exception of Lindes and Laird (2016, 2017), which can build structure from inputs in English and Spanish. The representation of the grammar is sufficient to generate and parse those sentences, and fit the experimental metrics. In doing this, the modeller makes choices about the distribution of the grammar across the model's architecture.

Full implementation of a computational algorithm such as left-corner parsing requires a stack, which is available within the Soar architecture for Hale (2014) and Lindes and Laird (2016, 2017), and which was implemented in Nengo by Stewart et al. (2014). This gives more flexibility to extend a grammar by adding new rules, but as Lindes and Laird (2016) point out, adding new rules is not necessarily a simple modelling task even with a stack. An unlimited stack may facilitate the processing of multiple centre-embedded clauses, which reliably cause processing difficulties in human subjects. However, arbitrarily limiting the depth of a stack excludes phenomena such as those reported by Fodor (2013), where prosody or pronominal arguments can improve the comperehensibility of multiple centre embedding.

Without a stack, modellers seeking to implement left-corner parsing must consider other strategies such as the specific phrase types of L&V05, and/or restrict the scope of their grammar fragment. L&V05 assume that the first nominal argument encountered is stipulated to be [Spec, IP], excluding sentences with a fronted topic such as *Peas, I like, but carrots, I hate.* Given the role of the [Spec, IP] assumption in generating initial structure on the first word, there is a considerable task involved in expanding the model grammar to allow sentence initial [Spec, CP] where the first word encountered is ambiguous between the two structures.

Generally, models are built to address a particular purpose, and the choices relating to grammar and architecture are made in response to that, rather than external criteria. Thus the models focusing on discourse processing (Anderson et al., 2001; Budiu and Anderson, 2004) included separate semantic and situational representations, whereas these are absent from L&V05. Conversely, the grammatical representation of L&V05, which focuses on structure building, is more flexible and more regularly aligned with X-bar theory than those in the earlier models. These architectural choices then have an impact on how modelling tasks are constructed. In the earlier models of Anderson et al. and Budiu and Anderson, referents are identified from the situational representation, whereas in the extensions of L&V05, resolution of anaphora takes place within syntactic structure, with the referent of an anaphor taken to be the head of a constituent in the syntactic representation, rather than some other form of representation. The levels of representation restrict the options for resolving ambiguity or computing constraints. The impact of the stereotypical gender of ‘surgeon’ on the time course of reflexive anaphor resolution depends not on linguistic knowledge, but upon real-world associations. Conflating these two into a syntactic process implicitly assumes that other interactions are secondary. From the perspective of LFG, constraints are expected to apply between levels of representation, and so the grammar offers a theoretical framework that allows these interactions to be modelled specifically.

4.3.4 Conclusion

In conclusion, it is possible to model elements of language processing successfully without a detailed consideration either of the specifics of grammar, or of the role of grammar in processing. However, if we want our model assumptions to be extendable to cover more general processing phenomena or grammatical constructions within a particular language, or if we want our model to be generalisable across typologically different languages, it is justifiable to require our assumptions about grammar to be more robust. A more robust set of assumptions does not mean that any model must seek to cover all of a language’s grammar, or to work cross-linguistically, but where we delimit a grammar fragment and set out the rules or conditions which apply to its parsing, we should take extensibility into account.

The nature of architectural assumptions is also important. ACT-R contains implicit assumptions about embodiment of cognition, with the hypothesised relationship between specific ACT-R modules and areas of the brain. Where models make assumptions about working memory capacity and configuration, such as the common assumption that structural elements are in some way typed, and that these types relate to elements specified by a language’s grammar, this may hide an implicit claim that there are embodied working memory differences between native speakers of different languages. This is a matter for empirical investigation, but a parsimonious account would start with the assumption that working memory operation is independent of a speaker's native language or languages.

Additional buffer capacity is a very common assumption when modelling language processing: of the ACT-R models reviewed, only that of Reitter et al. (2011) did not assume additional buffer capacity, and that model was considering production rather than comprehension. In being more robust about the grammar theory that underpins our assumptions, we can be clearer which elements of that theory are universal and which are language-specific. This then allows us to test hypotheses about the relationship between native language and working memory, if those are at issue, or to avoid implicit claims about the impact of native language if that is not at issue.

Chapter 5

Building a processing model from an incremental theory

Chapter 3 set out a cross-linguistic theory of incremental structural growth within LFG. That theory describes the set of structural growth options that are available when a new word is to be added to structure, sets out how the grammar constrains the set of options, identifies grammar-related factors that may affect the choice of an option from the constrained set, and as a result, sets the field for hypotheses about cognitive factors that may affect the choice of an option from the constrained set.

Chapter 4 then reviewed models of language processing, identifying how theories of grammar have been incorporated into them, and understanding their proposals about the impact of cognitive constraints on language processing. This included an introduction to the ACT-R cognitive environment. From this review, I identified a set of desiderata for a cognitive model of language processing based on incremental LFG.

In this chapter I bring together the incremental theory of grammar with the theory and practicalities of cognitive modelling to address the question: *how can the incremental theory of LFG be engineered into a cognitive model of processing, such that it is feasible to develop a model in the ACT-R cognitive environment, and such that the relationship between the model and other elements of the theory is clear and explicit?*

In Section 5.1, I place the modelling element of the thesis in the context of three theoretical areas. First, I set out assumptions about the language processing pathway that I am seeking to model, delimiting this within the wider context of language processing. Second, I relate this delimited task to the context of existing cognitive models of language processing, setting out the outputs that I will be using to evaluate models. Third, I delimit the scope of the modelling task within the grammar, specifying which phenomena and constructions are to be addressed.

In Section 5.2, I introduce the core processes and memory structures that will be used in developing the ACT-R models in Chapter 6 and show how they bring together the theoretical requirements from Chapter 3 and the modelling assumptions from Section 5.1.

And in Section 5.3, I discuss ACT-R modelling assumptions that are not constrained by the assumed processing pathway or the incremental grammar theory, and set out questions for more detailed consideration in Chapter 6, which describes the construction of models and the outputs of simulations.

5.1 The modelling task in the broader overall context

In this section I place my model in its theoretical context and specify the limits of my research both in terms of language processing, and in terms of theories of grammar. In Section 5.1.1, I set out the assumptions that I am making about language processing overall, based on the review of processing literature in Chapter 2. In Section 5.1.2, I map the incremental theory of LFG from Chapter 3 onto the assumed processing pathway. In Section 5.1.3, I define the scope of my model in terms of the processing theories discussed in Chapter 2, and in Section 5.1.4, I specify the grammar fragment and specific constructions to be tested in modelling.

5.1.1 Language comprehension in context

I assume that the aim of language comprehension is to infer the meaning of an utterance (Kuperberg and Jaeger, 2016). Following Brouwer et al. (2012, p. 136), I take the output of language comprehension to be "a mental representation of what is communicated (MRC for short)." From here on, I use *MRC* to refer to this mental representation.

The discussion in Chapter 2 showed that language comprehension tasks are carried out in many areas of the brain. Some of these areas seem to be specialised for language whereas other areas may also be involved in other cognitive functions. During the processing of an utterance, there is ongoing interaction between specialised language-processing areas, and between these areas and other parts of the brain. Accordingly, specifying the boundaries of a cognitive "language module" is not a simple task, if *module* is taken to mean a distinct element of cognition that is specialised to language processing, and within which language processing is restricted.

Authors working within the Good Enough Processing model (Ferreira and Patson, 2007) describe separate syntactic and semantic elements of language processing, but the boundary between these cognitive tasks is not explicitly defined: semantic cognition is the task of identifying meaning that is separate from building syntactic structure. However, this raises questions about the role of semantic compositionality and the detailed relationship between syntactic structure and meaning. The projection architecture of LFG allows the relationship between structure and meaning to be specified clearly, and so a division between "syntax" and "semantics" is not needed in defining the processing task. Instead, I propose to make a distinction between *compositional* and *inferential* processing in language comprehension.

Compositional processing might loosely be termed “bottom-up” processing: it is strictly based on the hearer’s knowledge of grammar and the contents of the grammatical lexical specification. This does not completely exclude predictions, but these should be based on grammatical requirements of a particular context rather than non-linguistic expectations. An example would be the string *Anka put the book* at the start of an utterance, where the valency of the verb *put* leads to a strong intuition that the next constituent will give information about the destination of the book.

By contrast, inferential processing uses the semantic content of the utterance together with real world knowledge. It may be constrained by grammatical structure, but is determined by the individual hearer, and is therefore predicted to vary significantly between hearers.

Thus the high-level architecture that I am assuming is shown in Figure 5.1.

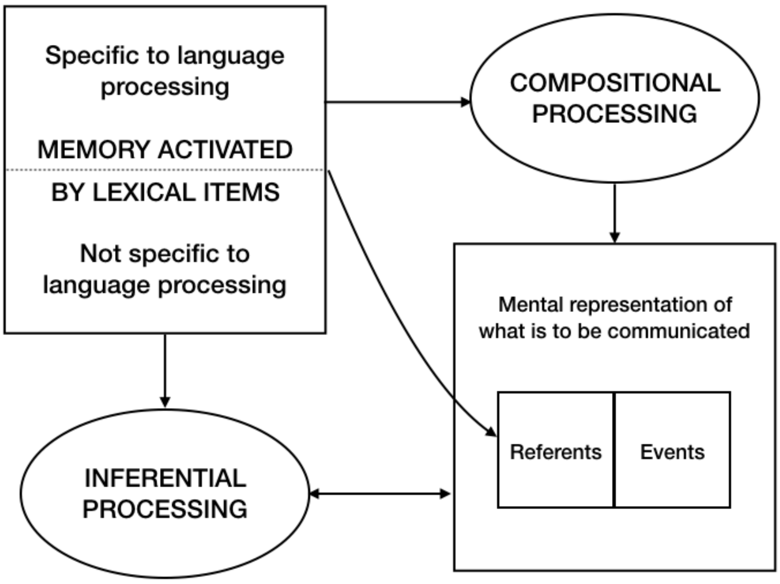


Figure 5.1: Language processing and wider cognition

Overall I assume the Cognitive Equilibrium Hypothesis proposed by Karimi and Ferreira (2015), that introduction of new information that is not consistent with the current cognitive state triggers processing until the new information can be integrated and a cognitive equilibrium is achieved. Within a sub-task, processing builds partial representations as soon as possible, which are then available to other sub-tasks. Thus although a later sub-task may be dependent on the output of an earlier sub-task, this output does not need to be completely specified in order for the later sub-task to commence.

I assume that the memory activated when a word is recognised includes both linguistically-specialised and non-specialised knowledge, and that both types of knowledge are necessarily activated. Compositional processing is the calculation of propositional meaning based on language-specialised knowledge, corresponding to the LFG calculations from the s-string to final meaning. Inferential processing is not grammatically determined in the same way, and I assume that non-specialised knowledge is therefore immediately available for processing outside the language-specialised areas, whereas linguistically-specialised knowledge must be processed before its contribution can be accessed by wider processing. Thus inferential computations may be faster than compositional computations.

With regard to the MRC, I make the further assumption that there is a direct link between the memory activated by linguistic input and the MRC, such that non-pronominal discourse referents are added to the MRC immediately, without requiring further processing. The outputs of inferential processing may contribute to or constrain the compositional processing, and may also contribute to the development of the overall MRC. At the same time, constraints imposed by grammatical structures may be taken into account in inferential computation, such as the on-line distinction between discourse referents and clausal arguments in Cloze tests described by Chow et al. (2015).

The task of the language-specialised areas of the brain is therefore to contribute to the overall MRC without necessarily being the sole source of the MRC's content. Other inferences may be added to the MRC before an utterance has finished, and a hearer may act on the basis of a partially-specified MRC before all linguistic input has been processed: this is in line with the predictions of the Good Enough Processing model.

5.1.1.1 Processes within language-specialised cognition

The box "Compositional Processing" in Figure 5.1 includes a number of processing subtasks that are specific to language. I assume that these tasks correspond to the 120-500ms period of the time course for processing described by Murphy et al. (2018), which was presented as Table 2.1 in Chapter 2 and is repeated here as Table 5.1. I assume that "Integration of syntax and semantics" corresponds to detailed construction of the discourse representation, and that "Interpretation" represents processing based on the contents of the discourse representation.

Table 5.1: Timing of language processing tasks (Murphy et al., 2018)

Time course	Tasks per Murphy et al.	
0-100ms	Stimuli perception	
ca. 150ms	Intelligibility	
120-200ms	Local structure building	
300-500ms	Semantic analysis	*possibly in parallel*
300-500ms	Syntactic analysis	
ca. 600ms	Integration of syntax & semantics	
> 600ms	Interpretation	

The terms used by Murphy et al. are challenging for grammar theories and cognitive modelling because they are broadly defined, with little detail of the boundaries between the tasks or of the processing content of individual tasks. For the purposes of this thesis, I assume that *local structure building* entails the assignment of head-dependent relationships to lexical content, that *semantic analysis* entails the assignment of relationships between events and their arguments, and the computation of a logical proposition based on that assignment. *Syntactic analysis* covers the calculation of syntactic relationships and constraints beyond local attachment, including the resolution of long-distance dependencies. This is a simplification: for example, it ignores any feature checking after the structure has been built initially, and it assumes that semantic analysis does not consider the contribution of modifiers to meaning. The specific tasks within compositional computation are shown in Figure 5.1.2.

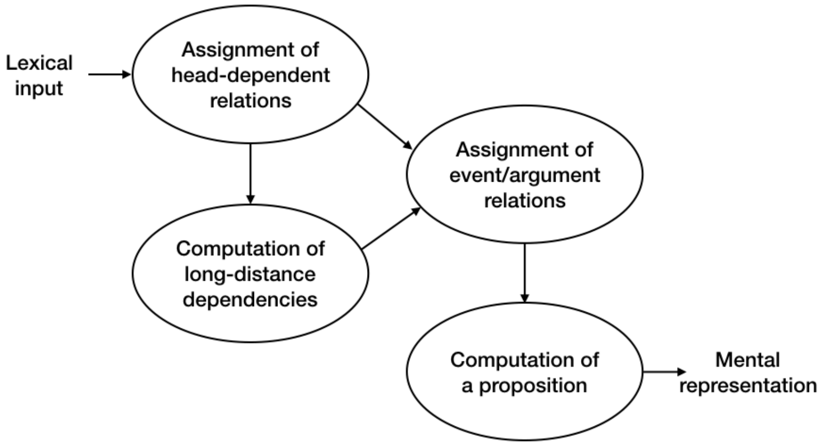


Figure 5.2: Specific tasks assumed within compositional processing

One consequence of my assumptions on the language-specialised sub-tasks and the Cognitive Equilibrium Hypothesis is that the semantic and syntactic analyses take place in parallel. Thus a partially-specified semantic representation may be augmented with further content once a long-distance dependency is resolved. Another consequence is that propositional content is added to the MRC as soon as it has been computed, even if that computation results in a proposition with many variables.

5.1.2 Mapping LFG onto the processing pathway

LFG assumes distinct levels of representation for constituent, functional, and semantic structure with projection functions determining an ordered relationship between the levels. All levels of representation are assumed to be present simultaneously, and there is no implicit claim about the time sequence of processing in the brain. The modular architecture potentially allows different elements of processing to be treated separately, but that requires a correspondence between the levels of representation and specific processing tasks.

In Figure 5.3, I propose a relationship between the elements of compositional processing and LFG levels of representation. There are two clear areas of mismatch. The first is the assignment of head-dependent relationships which, as discussed in Section 3.3, involves elements of both c- and f-structure, with constraints from both levels of representation coming into play when determining an attachment site. The second is the computation of

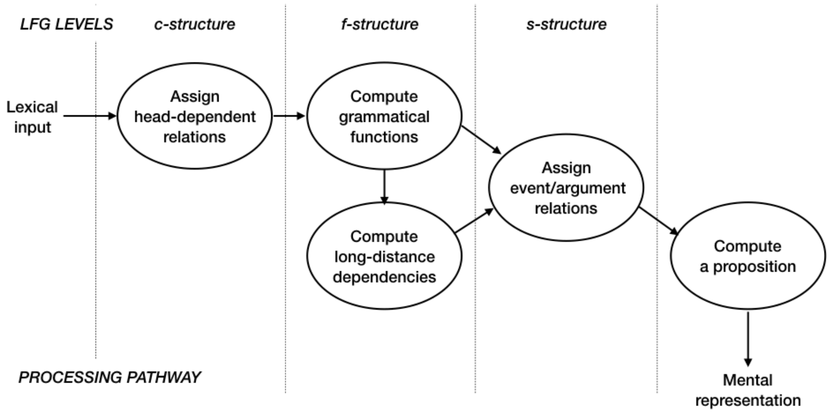


Figure 5.3: Correspondences between LFG and the assumed processing pathway

the σ projection function between f-structure and s-structure, which requires input from two distinct processing tasks: assignment of head-dependent relations and computation of long-distance dependencies.

From an LFG perspective, these mismatches may not be problematic if the final mental representation of head-dependent relations allows both c- and f-structures to be retrieved, and if the processes that contribute to the σ-function produce outcomes that are congruent. In Section 5.2.4, I propose a single syntactic representation for the purposes of modelling incremental processing, that incorporates information from f-structure with combinatorial and ordering constraints derived from c-structure. I also assume that the assignment of hierarchical relations, grammatical functions, and the resolution of long-distance dependencies take place in the same cycle. This is a simplification from the model illustrated in Figures and 5.3, and a consequent assumption is that there is no potential for parallel input to s-structure. This assumption meets the scope of this project in testing the incremental theory, but must be revisited if the project is extended to consider further LFG levels of representation.

5.1.2.1 Hierarchies and parallelism

The relationship between LFG levels of representation and the processing pathway set out in Figure 5.3 contains assumptions that are inherently contradictory. The proposed processing pathway is sequential, with individual tasks on that pathway taking time to complete. Even if a task requires only one ACT-R production to fire, there is still a 50ms difference bewteen input and output. By contrast, an LFG analysis is declarative: all elements of the analysis are available simultaneously and mutually constrain each other.

Delimiting the project to initial structure building reduces the tension between the contradictory assumptions: only one step in the pathway is being considered, and so any issues raised by levels of representation being out of step with each other are moot. The model also assumes that one word is processed at a time. Because new input is not attended until the previous input has been processed, the buffer state of the model is fixed at the point that the new input must be processed.

These assumptions do not stand up to empirical data on processing, nor are they in line with theories of hierarchical processing (e.g. Lupyan and Clark, 2015). The empirically-derived processing time course described by Murphy et al. (2018) shown in Table 5.1 assumes that each word takes >600 ms to be integrated into the MRC, whereas in normal speech a new word may be encountered every ∼300ms, and written language may be processed even more quickly. The logical consequence of this is that elements of the processing pathway are operating in parallel, with syntactic representations for word$_{n+1}$ being computed before word$_n$ has been integrated into the discourse structure. Constraints may still operate between processing tasks that correspond to LFG levels of representation, but constraints that operate backwards through the LFG projection architecture (e.g. s-structural constraints on f-structure) are not available instantly in this model.

Lupyan and Clark's hierarchical processing model assumes that more abstract processing tiers set expectations of, and constraints on, lower tiers, and that "residual" processing tasks that cannot be resolved at a particular tier are passed up to a higher level for resolution. However, once a residual issue has been resolved, although it may constrain future processing, it is not clear that any incomplete representations at a lower tier are revisited and corrected. Indeed, a lack of thorough reanalysis at all tiers is in line with the persistence of meanings derived from the processing of garden path sentences (Christianson et al., 2001).

Future exploration of this relationship, outside the scope of this project, will allow predictions about the effects of the delayed availability of constraints, and the impact of incomplete reanalysis, to be generated and tested. Authors working within LFG have made proposals about where a particular constraint is most parsimoniously sited (e.g. Asudeh and Giorgolo, 2012). Future models should allow the impact of a constraint appearing earlier or later in the projection architecture to be compared with empirical evidence, and hence contribute to the development of LFG theory.

5.1.3 Delimiting the modelling task

The purpose of this project is to test the hypothesis that it is possible to build a universal syntactic representation of (written) language input from two typologically different languages using a single production architecture and language-specific lexicons that contain information on the constraints that apply to combinations of phrasal categories, and constraints that apply to the order in which grammatical functions are provided.

The aim of the model is also to test the following specific predictions arising from the incremental theory of LFG presented in Chapter 3.

(i) Processing a sentence produces a single structural representation, with only one option being selected if more than one is available.

(ii) Where there is syntactic ambiguity, multiple runs of the model should produce a distribution of output structures that matches the distribution of human subjects.

(iii) Structure building is monotonic, and so reanalysis of a previously-built structure is a separate processing task to structure-building: this should correlate with observed garden path phenomena.

(iv) Attachment choices, including structural attachment preferences such as Minimal Attachment and Late Closure (Frazier, 1987), are context specific and so similar strings may be processed differently in different contexts.

I assume “maximally incremental” processing, which requires that the information from each word is processed as soon as the word is perceived, and that this information is incorporated immediately into the representation of an incrementally-growing utterance.

Because the modelling is undertaken in ACT-R, model time-course data is available for the processing cycles. However, this thesis makes no claims about matching empirical time-courses for processing specific constructions. Accordingly the model time-course data does not form part of the hypothesis testing, and is used for illustrative purposes.

In constructing a model in ACT-R, the modeller must make assumptions are made about the functional division between declarative and procedural memory, the size and function of any additional capacity, the structure of memory chunks and search cues, the level of central control of the processing task, and the size and role of any additional buffer capacity.

At the moment, the wider process hypothesis about the internal stages of language processing, and the interface with wider cognition is not being tested. The ACT-R environment will allow models to be constructed to do that, but if they are to allow a thorough discussion of the role of grammar in processing, it is necessary to check first of all that the constraints of the grammar are being faithfully reproduced in the processing model.

5.1.4 The grammatical scope of the model

The theory elaborated in Chapter 3 claims that lexically-specified constraints together with a universal set of possible combinatorial operations are sufficient to allow the building of syntactic structure incrementally. The aim of the modelling is to test those claims looking at two typologically different languages. The focus of modelling is accordingly on the accuracy of structures that are built during processing compared to structure that are congruent with accuracy or meaning judgements reported by human subjects, rather than on the time courses of particular phenomena. I propose to test this by building simple monoclausal sentences, and complex sentences with clauses subordinate to nouns or to verbs.

The test sentences will include sentence pairs where the same string of words produces a different structure in a different sentential context, including the presence or absence of a garden-path effect. The repair of ungrammatical garden paths is not included in the scope of the model, as this is assumed to use higher-level cognition.

5.2 Core elements of the model

In this section I motivate my assumptions around three core aspects of the cognitive model: the number and type of linguistic representations being calculated; the processing cycle that is assumed when structure is being built incrementally; and the specific elements of syntactic representation that must be present to justify the claim that this is a model of LFG. I begin by setting out the case for three tiers of linguistic representation in the model (of which I will be testing two) which include the *F-representation*, a single representation of syntactic structure, before describing the processing algorithm. I then specify the contents of the F-representation and show that it allows an LFG analysis to be retrieved, and that it allows an ACT-R model to function.

5.2.1 Representing syntactic structure, propositional content, and the discourse

The processing pathways depicted in Figures 5.1 and 5.1.2 all culminate in the MRC, whose content is derived from compositional and inferential processing. Within compositional and inferential processing, propositional output to the MRC is derived from a representation of argument structure whose input is from the assignment of head-dependent relations. The most parsimonious account of compositional processing would have a single representation without requiring an intermediate representation of syntactic structure. However, I argue that there are theoretical and empirical grounds for assuming two representations, syntactic and semantic, within compositional processing.

From the perspective of grammar theories, there are many potential mappings from syntactic structure to thematic role, whether syntax is defined in terms of positions within some kind of syntactic tree or in terms of grammatical functions. Theories such as Dynamic Syntax (Cann et al., 2005) assume a single representation in terms of a semantic tree whose nodes represent type-theoretical categories. However, the syntactic structure of an utterance is not recoverable from the tree: instead the syntax imposes lexical and structural constraints on the order in which the semantic tree is built. Thus although there is only one output representation, the theory still distinguishes syntax and semantics. Elsewhere, the assumption of separate representations is common, requiring proposals about the mapping relationship between the two.

Empirical evidence comes from neurolinguistic examination of brain activation, which was discussed in Chapter 2. Friederici (2011) describes distinct and separate areas of activity localised for syntactic and semantic processing, with activity in one area not necessarily entailing activity in another. Given that processing in these areas may also proceed in parallel, it may be more parsimonious to assume two distinct representations associated with the different brain regions, rather than a single but more complex representation that captures the possible variations in processing activity.

Analogous to the models of Anderson et al. (2001) and Budiu and Anderson (2004), I assume that the syntactic, semantic, and discourse representations of the language input are distinct types of representation computed during processing. Although they are based on LFG structures, they may have different characteristics. Accordingly I will refer to them as “representations” rather than “structures”. The three types are:

(i) syntactic, corresponding to LFG c- and f-structure;

(ii) semantic, corresponding to LFG s-structure;

(iii) discourse, containing the computed output of language processing, which may also include items added by wider cognition.

In line with the proposed model of language processing depicted in Figure 5.1, I assume that the syntactic and semantic representations are specialised to language and cannot be accessed directly by wider cognition. However, their outputs in terms of path descriptions and thematic roles are captured in the discourse representation and so may be used in wider cognitive processes. Also, the computation of the syntactic and semantic representation makes use of the content of the discourse representation and, as such, may include elements of that representation that were added by wider cognition rather than by the language module. For the purposes of testing the hypothesis I will be modelling only the construction of syntactic and discourse representations.

5.2.2 A single syntactic representation

The closest correspondence between an ACT-R cognitive model and an incremental theory of LFG would entail building a c-structure representation and projecting an f-structure representation from this. However, this approach raises theoretical and practical issues. Some of these arise from LFG assumptions including those related to cross-language variability and universal structure, and others arise because of interactions between LFG and the structural assumptions of ACT-R. I will consider these in turn.

5.2.2.1 The impact of LFG assumptions on model design

The role of c-structure in mediating the relationship between structure and meaning varies greatly between languages. Bresnan et al. (2016) describe the characteristics of exocentricity and endocentricity in c-structure, noting that in many languages it is possible to observe both strategies in their phrase structure constraints. They also note the varying contribution of lexical information to the assignment of grammatical functions, with some languages showing very little agreement or case morphology, and others showing rich morphological marking. Where marking is seen, this may be on heads, on dependents, or on both. This variation has implications for the modelling of c-structure in memory, both for the shape of the memory representations and the capacity needed to generate and maintain these representations.

The cognitive model representations of syntax trees discussed in Chapter 4 (e.g. Lewis and Vasishth, 2005; Hale, 2014) have based their memory representation on the tree node. Because the nature of a node is determined by theory-internal assumptions about levels of projection and the number of branches permitted at a node, these assumptions also influence the shape of memory representations. X-bar theory (Chomsky, 1968; Jackendoff, 1977) is commonly assumed in LFG, although authors differ in their assumptions about whether a phrasal head from a particular category X^0 always projects an intermediate X' projection as well as a maximal projection X''/XP, and about the availability of non-projecting categories $\widehat{\mathrm{X}}$ (Toivonen, 2003). Adding a word and its projections into the representation of a syntax tree may require the creation of multiple intermediate nodes: in the model proposed by Lewis and Vasishth this required the addition of twelve hidden buffers to ACT-R structure.

The variation in permitted branching also has an impact in modelling. In languages where endocentricity and configurationality are theoretically important in determining grammatical function, nodes (and therefore the corresponding memory representations) are likely to be restricted to binary or ternary branching. However, in exocentric, highly non-configurational languages there may be no restriction on the number of branches to a node (and therefore to the shape of the corresponding memory representation). If we are seeking to build a model that is applicable across natural languages, this variation in the shape of our basic memory representation is undesirable.

In Section 5.1.2.1 I discussed some of the challenges arising from the tension between a sequential processing pathway and a declarative theoretical framework. These issues also potentially apply when calculating c-structure and f-structure, LFG's two levels of syntactic representation. The mutual constraints between c-structure and f-structure must be faithfully reproduced if a model is to generate a meaningful syntactic representation,

5.2.2.2 A single representation for modelling purposes

The underspecification and ambiguity that arises during incremental structure-building is expressed differently in terms of c- and f-structure. In general, f-structure can capture underspecification more economically than c-structure. For example, Figure 5.4 is an ATN representing the English rules for phrase structure projected by categories C, I and S, taken from Appendix A[1], considering the alternatives when a right-hand side category is present or absent. For the moment, adjunction is ignored.

Discounting the potential for recurrence of a CP, there are eighteen distinct pathways

---

[1] Falk's treatment of clausal subjects introduced by a prepositional phrase, e.g. *For the modeller to omit this construction would be permissible* is not included.

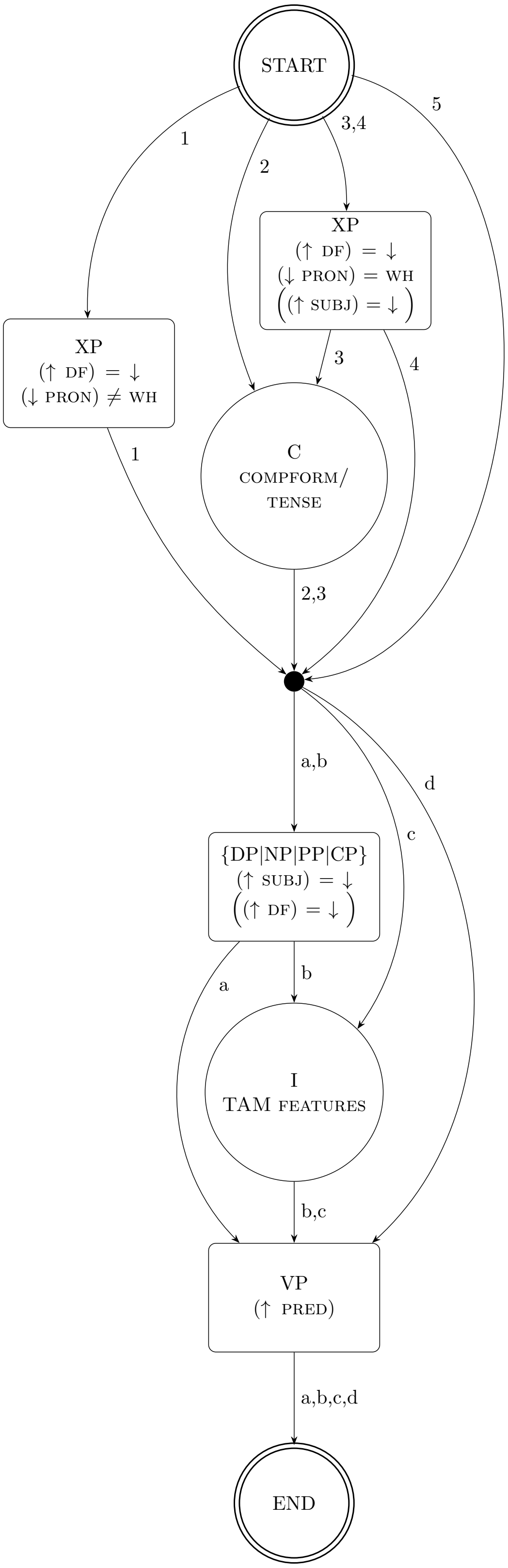


Figure 5.4: Pathways through the grammar's CP, IP, and S phrase-structure rules

through the network[2]. These are shown in Table 5.2 with example sentences, having been generated by following each possible path in turn from the start to the end, working

---

[2] Pathways 1c, 1d are effectively identical to 5a, 5b.

through the options at each node from left to right.

Table 5.2: Clauses generated from Figure 5.4 that project the f-structure in (179)

		DF (SUBJ†)	COMP-FORM (TENSE)	SUBJ	COMP-FORM	TAM FEATURES	PRED	Other GFs
1a		Cakes,		David			bakes,	*but pies, he…*
1b		Cakes,		David		will	bake	*but pies, he…*
2a			Does	David			bake	*cakes?*
2b	*I think*		that	David		is	baking	*cakes today.*
2c	*He seems*				to	be	baking	*cakes.*
2d	*He wants*				to		bake	*cakes.*
3a		What	is	David			baking?	
3b		When	will	David		have	baked	*them all?*
3c		Who†	will			be	eating	*them all?*
3d		Who†	will				want	*more?*
4a	*the cakes*		that	David			bakes	
4b	*the cakes*		that	David		is	baking	
4c	*the man*	who†					bakes	*cakes...*
4d	*the man*	who†				is	baking	*cakes...*
5a				David			bakes	*cakes.*
5b				David		is	baking	*cakes.*
5c						Do	bake	*cakes!*
5d							Bake	*cakes!*

The examples in Table 5.2 include matrix clauses in declarative (1a, 1b, 5a, 5b), interrogative (2a, 3a, 3b, 3c, 3d) and imperative (5c, 5d) moods, as well as embedded clauses (2b, 2c, 2d, 4a, 4b, 4c, 4d) and clauses with a fronted topic (1a, 1b) . However, all are captured by the f-structure shown in (179), and where a grammatical function is lexically contributed, the order of contribution is ① $\prec$②$\prec$③. Furthermore, there is no case where a DF is provided without a SUBJ also being present: where a governed grammatical function immediately precedes the constituent that provides the PRED value, it is the subject. The f-structure in (179) is a simplification and other computation is needed to ensure that the full meaning is represented, but building a representation based on f-structure allows this work to be separated from structure-building without having to make early assumptions about how an utterance will progress.

(179)
$$\left[\begin{array}{llll} \Big(& ① \quad \textsc{df} & \left[\textsc{pred}\quad \text{`pro' / `cakes'}\right] & \Big) \\ \Big(& ② \quad \textsc{subj} & \left[\textsc{pred}\quad \text{`David'}\right] & \Big) \\ & ③ \quad \textsc{pred} & \text{`bake}\left\langle \textsc{subj},\ \textsc{obj}\right\rangle\text{'} & \\ & \quad\vdots & \quad\vdots & \end{array}\right]$$

On the basis of this evidence, I assume that it is preferable in modelling terms to generate a single syntactic representation, rather than attempt to build parallel c- and f-structures. In the context of a cross-linguistic model, the universal f-structure is a more promising basis for this structure than language-specific c-structures. However, c-structure cannot be ignored altogether. Its role in describing dominance and precedence relationships between constituents of different categories, and its contribution to the calculation of f-precedence and the CAT predicate, mean that sufficient c-structure information must be available at choice points. Furthermore, if an LFG analysis is to be recoverable from the output of the model, there must be sufficient information from c-structure available to allow this to take place.

This raises the question of exactly what information is required to support decision-taking at choice points, and to allow LFG analyses to be recovered. To answer these questions, I will consider the structure-building process and then identify how the LFG c-structure and f-structure constraints might be applied within that process.

In section 5.2.4 I show how these elements of the syntax can be captured in a structure that I term an *F-representation*. This includes elements of f-structure, the f-structure reflexes of c-structure constraints as described in Section 3.3, and elements which are purely c-structural such as category constraints. F-representations also have characteristics of c-structure, such as an ordering of some elements, which are not present in f-structure. The elements and characteristics of F-representations interact with lexical information and word order in a string to determine the set of choices available when adding a word to an incrementally-growing representation.

5.2.3 The model process for building structure

I assume a high-level process for building structure in line with the language processing pathway described by Murphy et al. (2018) presented in Table 5.1. This has the following steps.

1. Attend to a word

2. Retrieve and select the best match lexical entry

3. Add a new referent to the discourse if indicated by the lexical entry

4. Choose an attachment site

5. Add the lexical information into the existing structure

6. Take account of the revised structure's impact on long-distance dependencies

7. Compute the new set of potential attachment sites

This process does not include computation of propositional content or the addition of propositional information to the discourse.

5.2.3.1 Attend to a word

The model must be able to access one word at a time. This is not affected by any theoretical assumptions from LFG, but requires modeller decisions relating to ACT-R. In the ACT-R modelling I assume that the model determines when the next word is accessed, adapting the method of Lewis and Vasishth (2005) and Engelmann (2016) which uses the ACT-R Visual module to simulate eye-tracking experimental methods. I have not considered alternative approaches that would involve the ACT-R Audio module to interact with a signal that proceeds at a rate independent of the model's processing speed, or models that would use the ACT-R Motor module to simulate self-paced reading experiments.

This method requires an assumption about the point at which the model is ready to proceed to the next word: in the models reported in Chapter 6, I assume that this does not take place until the end of the model processing cycle.

5.2.3.2 Retrieve and select the best match lexical entry

Evidence from language processing experiments suggest that context can be factor in the on-line resolution of lexical ambiguities including category (e.g. Duffy et al., 1988). In an incremental theory of LFG, prior context is available from the propositional representation as well as constraints within the set of available attachment sites. However, for the purposes of this thesis, it is assumed that lexical ambiguities have already been resolved.

5.2.3.3 Add a new referent to the discourse if indicated by the lexical entry

I assume that a semantically weighty PRED is one that contributes a new discourse referent, and that this takes place early in the processing cycle. I assume that pronouns do not contribute new discourse referents, but instead trigger an initial search for an association with an antecedent. The early contribution of a discourse referent may reflect co-activation of language-specific and general knowledge when a lexical item is retrieved. From an LFG perspective, this does not require any structural information, merely a lexical specification that contains a PRED value other than 'pro'. The evidence from Chow et al. (2015) is that computations of Cloze items are sensitive to syntactic structure as well as semantic meaning. This may entail that both syntactic and discourse representation are used in the computations, or that a discourse representation includes some kind of syntactic computation such as the path between f-structures (described in Chapter 3), which would be dependent on the f-structure containing the PRED value having been added to the structure. I assume that syntactic attachment is not necessary in order for a new referent to be added to the discourse. The question of whether syntactic representations are recovered alongside discourse representations, or whether a discourse representation includes some kind of syntactic index, is left for future work. I also do not consider the computations that would be necessary to map multiple syntactically weighty descriptions (e.g. *An elephant... The grey giant...*) to the same referent.

5.2.3.4 Choose an attachment site

The set of candidate attachment sites is computed at step 7 at the end of every processing cycle and is thus available after the first word in the utterance. There are at least two computations inherent in choosing an attachment site. One is a potential restriction of the set of candidate sites if the grammatical function or categorial constraints that a candidate site imposes are incompatible with the constraints contained in the lexical specification. Another is the ranking of potential sites against processing constraints or preferences (e.g. "add a governed grammatical function in preference to an adjunct"). Beyond these, context or structural priming factors may also have an impact. In the model described in Chapter 6, I consider only the first two computations.

From an LFG perspective, neither the candidate attachment sites nor the lexical specification are restricted to outside-in or inside-out constraints: in Section 3.3.2 it was shown that f-structure might be added outside, inside, or between existing f-structures during incremental growth. Accordingly, any inside-out and outside-in category and grammatical function constraints must be available for computation. For ACT-R modelling purposes, I assume that the constraints contained in the lexical specification are used as search cues within the set of candidate attachment sites.

The process must also include the possibility of introducing an underspecified F-representation: it may be that an underspecified F-representation is introduced in preference to direct attachment. Modelling allows hypotheses about the factors influencing these preferences to be tested.

5.2.3.5 Add the lexical information into the existing representation

Once an attachment site has been selected, information must be copied from the lexical specification into the structure. This requires no LFG-specific assumptions. The model in Chapter 6 explores how the ACT-R production architecture can effect the range of attachment possibilities identified in Section 3.3.2.

5.2.3.6 Take account of the revised structure's impact on long-distance dependencies

A lexical specification may introduce a new long-distance dependency or it may indicate that a long-distance dependency is closed. Alternatively, it may indicate that an already-open long-distance dependency may not be closed at that point.

Chapter 2 discussed the evidence that processing of written English slows down at points in an utterance where an open long-distance dependency may be closed. Some accounts see the slowdown as relating to an active attempt to resolve the dependency, so that there is automatic activation of the dependency whenever a suitable site is found; in other accounts the long-distance dependency is always open for attachment and the amount of competition between the LDD and the word currently being processed governs the retrieval time. Both of these strategies can be modelled in ACT-R.

5.2.3.7 Calculate the new candidate set of attachment sites

In Section 3.3 the role of the set R of nodes along the right edge of c-structure in constraining attachment was discussed. In terms of F-representations, the candidate set of attachment sites is a subset of the pathway from the outermost F-representation to the F-representation where lexical information was just added. Specifically, it is those F-representations that lie on the path from the innermost F-representation without a PRED value to the F-representation where lexical information was just added. Within ACT-R, calculating this from scratch after each attachment would require the retrieval of each chunk along the pathway from the attachment site to the outermost possible candidate site, which would be reflected either in additional processing time or in additional buffer capacity to maintain the structure actively.

An alternative approach would be for each F-representation to include a record of its pathway to the outermost f-structure at the point of attachment. This is easily calculated if the pathway is represented as a list: the pathway for the new F-representation is added to the end of the pathway list of its immediately containing F-representation. The set of candidate attachment sites can then be calculated by working backwards through the list until either the outermost F-representation is reached, or until an F-representation without a PRED value is reached. All of the F-representations before this point in the list are excluded from the candidate set.

5.2.3.8 Information needed at choice points

In summary, to add a new word into the structure, we need to know the following.

(i) The set of candidate attachment sites.

(ii) Outside-in and inside-out constraints present within the candidate attachment sites and the lexical specification:

  - category;
  - category constraints on the mother;
  - governed grammatical functions;
  - constraints on the GF that a word may provide;
  - ordering constraints on governed GFs;
  - ordering constraints in relation to the mother's PRED.

(iii) Attachment preferences (e.g. add a governed GF in preference to an adjunct).

(iv) To compute long-distance dependencies:

  - the category (and potentially also the GF) of any LDD;
  - the starting point of an LDD (the point at which it becomes clear that an LDD is present);
  - the pathway from the starting point of an LDD to its anchor point (Dalrymple and King, 2013) (the point at which the LDD is resolved).

The information in (i) and (iv) relates to the overall context and so from an ACT-R perspective, could be maintained in separate buffers or as slot values in the Goal buffer. The information from (ii) relates to lexical specifications and individual F-representations within the overall structure, to be used as search cues or to hold structural constraints. In ACT-R terms, this suggests that the information must be included in chunk specifications.

If the information from (iii) is valid across languages, it could be included in ACT-R as part of the production set in procedural memory. However, if it is language-specific information, it may be possible to encode it in ACT-R as a language-specific variable accessed by productions, or within individual lexical specifications, rather than requiring language-specific productions alongside universal productions.

In Sections 5.2.4 and 5.2.5 I specify the content of the two types of declarative memory used in modelling LFG: F-representations and lexical specifications. I then continue in Sections 5.2.6 and 5.2.7 to discuss specific issues regarding long-distance dependencies and valency alternations.

5.2.4 The content of F-representations

Formally, the F-representation is a function whose domain is a partially ordered set of attributes that are taken or derived from f- and c-structure, and whose range is a set of values that may be atoms, F-representations, or sets of atoms or F-representations. The attributes required in an F-representation are shown in Table 5.3.

All of the attributes of the F-representation are provided by lexical specifications, with the exception of the attribute MA, which is added at the time of attachment. Lexical specifications contain additional information which allows the suitability of an attachment site to be assessed, but which is not used once an F-representation has been attached. This information is presented in Table 5.4.

5.2.4.1 Elements derived from f-structure

PRED The PRED value is supplied by the lexical specification, and must be unique, analogously to the PRED value of an f-structure.

Governed GFs Where a PRED governs grammatical functions, these must be included in the F-representation, analogously to f-structure. For ACT-R, this suggests that there

Table 5.3: Relationship of F-representation attributes to f- and c-structure

Derived from f-structure	
PRED	The semantic index, equivalent to f PRED
Governed GFs	Set of attributes representing grammatical functions obligatorily or optionally governed by the PRED
ADJ	Modifiers of the PRED
FEATURES	Syntactic and semantic features (e.g. number, person, gender) carried by the PRED
MA	The 'mother' F-representation: the immediately-containing F-representation at the time that this F-representation is added
Derived from c-structure	
CAT	The category of the word providing the PRED value
GF ordering constraints	The order in which grammatical functions are expected to be contributed to the F-representation.

Table 5.4: Additional contents of lexical specifications derived from f- and c-structure

Derived from f-structure	
MA.GF	A specification of the grammatical functions that the F-representation can provide in relation to its mother
Derived from c-structure	
MA.CAT	The category of the word supplying the PRED value for the mother F-representation.
MA.PRED	The status of the mother's PRED value at the point when this F-representation is added as a grammatical function.

is potentially a slot for each grammatical function, which can be used to hold the requirement or option for the function to be present, and to hold the chunk corresponding to the F-representation that provides the grammatical function. Issues relating to valency alternations and optional grammatical functions are discussed in more detail in Section 5.2.7.

ADJ I assume that any F-representation may contain adjuncts: this attribute represents the set of modifiers that are attached.

FEATURES Non-categorial syntactic and semantic features are captured in f-structure, and correspondingly will be included in F-representations.

MA This attribute holds the value of the mother F-representation. For F-representations, analogously to f-structures as described in Section 3.3.2, incremental growth can result in mother relationships between f-structures being disrupted, but the containment relation between f-structures does not change during incremental growth.

In an ACT-R representation, updating the MA value where the mother relation changes would require model time to retrieve and amend chunks, and/or additional buffer capacity. For the models in this thesis, I am assuming that this has no impact on processing, and so the MA attribute will hold the value of the mother F-representation at the time of attachment, representing a containment relationship.

5.2.4.2 Elements derived from c-structure

These elements represent the combinatorial constraints on phrase attachment

CAT Following Dalrymple (2017), I assume that category labels are composed of feature sets. In an F-representation F, the CAT attribute is the category feature set of the lexical head that contributes the PRED value to the corresponding f-structure f. As discussed in Section 3.1.5.5, this feature set is shared by all of the c-structure nodes in $\phi^{-1}(f)$. Nodes representing different levels of phrasal projection or functional phrases will have different values for bar and functional features, but will share the same category feature set.

GF ordering constraints This corresponds to the precedence order of grammatical functions derived from phrase-structure rules in Section 3.3.2.8. Formally, the relation in (180) holds between an f-structure f and its corresponding F-representation F.

(180) $f\ \textsc{gf}_1 \prec_f f\ \textsc{gf}_2 \Rightarrow F\ \textsc{gf}_1 \prec F\ \textsc{gf}_2$

An important distinction is between GFs that obligatorily precede the PRED and GFs that obligatorily follow. A similar distinction is relevant for adjuncts. I return to questions about how best to model ordering constraints in Chapter 6.

5.2.5 Consequences for lexical specifications

MA.GF The GF that a word may provide in relation to a head can be constrained bottom-up in two ways. The first relates to an explicit lexical specification, for example, *sakwalul* 'apple.OBJ' in Korean or *he* '3.SG.M.SBJ' in English. Where a GF is lexically specified, this constrains the GF of the F-representation at its anchor point, notwithstanding that this F-representation may fulfil more than one GF.

The second way to constrain the available GF arises from phrase-structure rules. Thus in the grammar of English used for this thesis (Appendix A), category N may provide SUBJ, OBJ, $\textsc{obj}_\theta$ or DF to a head, but may not provide $\textsc{obl}_\theta$, COMP, XCOMP, XADJ or ADJ. I will return in Section 5.2.7 to consider how this might effectively be captured both theoretically and also practically in terms of ACT-R chunk specifications.

MA.CAT This is the CAT value of the immediately containing F-representation f, and corresponds to the category feature set of the node to which the topmost node in $\phi^{-1}(f)$ attaches. Formally, the definition is given in (181): the equation says that the MA.CAT value of an F-representation F is the value of the labelling function of the mother of a particular node, where that node projects the corresponding f-structure f, but its mother node does not project f-structure f.

(181) F MA.CAT $= \lambda(\mathrm{M}(\bullet_x)) : \bullet_x \in \phi^{-1}(f) \wedge (\mathrm{M}(\bullet_x)) \notin \phi^{-1}(f)$

The MA.CAT attribute represents the members of the $X^m_{m \prec \mathrm{X}}$ and $X^m_{\mathrm{X} \prec m}$ sets defined in Section 3.3.1.2. It is contributed to the F-representation by the specification of the lexical head and may be a fully-specified or an underspecified feature set.

MA.PRED This corresponds to the phrase-structure constraint on whether a dependent must appear before or after the head of the phrase to which it attaches. If F MA.CAT = Y, then the phrase structure constraint relates to the members of the sets $Y^d_{d \prec \mathrm{Y}}$ and $Y^d_{\mathrm{Y} \prec d}$ defined in Section 3.3.1.2.

5.2.5.1 Constructing lexical specifications

Lexical specifications in memory need to carry sufficient information to allow F-representations to be populated when a word is added to the structure. This corresponds to a combination of LFG lexical entries and additional category-based templates so that the full implications of category phrase-structure constraints can be included. Consider the lexical specification for *kissed* given at (35a), repeated here as (182).

(182) *kissed* V $(\uparrow \text{PRED}) =$ 'kiss'
$(\uparrow \text{TENSE}) = \text{PST}$

$(\uparrow\ \text{SUBJ})_\sigma = (\uparrow_\sigma\ \text{ARG1})$
$(\uparrow\ \text{OBJ})_\sigma = (\uparrow_\sigma\ \text{ARG2})$

$\lambda y \lambda x \lambda e.kiss(e) \wedge agent(e) = x \wedge patient(e) = y :$
$(\uparrow_\sigma\ \text{ARG2}) \multimap (\uparrow_\sigma\ \text{ARG1}) \multimap (\uparrow_\sigma\ \text{EVENT}) \multimap \uparrow_\sigma$

The category V can be treated as a language-specific template @V that in English contributes the constraints in (183).

(183)

CAT	[ADJ −, ADV −, N −, P −, V +]
MA.CAT	
MA.GF	$\{\perp \vert \text{COMP} \vert \text{XCOMP} \vert \text{ADJ} \vert \text{DF}\}$
MA.PRED	+
GF.PREC	SUBJ
GF.POST	$\text{OBJ} \prec \text{OBJ}_\theta \prec \text{XCOMP} \prec \text{OBL}_\theta \prec \text{COMP}$
ADV.PREC	
ADV.POST	

Practical approaches to modelling this in ACT-R are discussed further in Chapter 6.

5.2.6 Long-distance dependencies

Clifton and Frazier (1989) propose the Active Filler Hypothesis, which assumes that while a long-distance dependency is unresolved, the parser will attempt to resolve it at each point in the parse where a resolution is possible, even if that resolution is revoked by subsequent content. Wagers (2013) suggests that while the search to resolve a long-distance dependency is ongoing, not only is the knowledge of an unresolved long-distance dependency actively maintenance in memory, but that knowledge includes some high-level information (e.g. category) about the nature of that dependency. However, more detailed information about the dependency has to be retrieved from memory. For ACT-R modelling purposes, active maintenance of knowledge suggests either a dedicated buffer or a chunk in the Goal buffer. Modelling may then explore what kind of mechanisms might be involved in resolving the long-distance dependency, including the memory retrievals and links between memory chunks necessary for the LFG analysis to be recoverable subsequently.

5.2.7 Ambiguity and underspecification

A cross-linguistic model must be capable of dealing with lexical and structural strategies for assigning grammatical function, and for inside-out and outside-in expression of attachment constraints. The crucial condition for attachment is that there must be no mismatch between constraints imposed from a directly contained F-representation and those imposed from the directly containing F-representation. However, either or both of the F-representations may be fully specified, partially specified, or unspecified.

The F-representation must be capable of dealing with valency alternations., e.g. English 'know ⟨SUBJ, {OBJ|COMP}⟩'), and optionality of grammatical functions, e.g. English 'eat ⟨SUBJ, (OBJ)⟩'. As discussed in Chapter 3, several methods have been proposed in LFG for addressing this. Some proposals (e.g. competing lexical entries) implicitly make predictions for processing that can be empirically tested. The approach used in F-representations must be congruent with LFG proposals, and must also be congruent with the constraints on MA.GF discussed above.

As discussed in Chapters 2 and 3, the same word sequence in different contexts results in different processing loads which may be linked to the need to reanalyse a grammatical function or an attachment. For the purposes of this thesis, the main example to consider is the valency alternation OBJ ↔ COMP that licenses the reanalysis of *f* OBJ to *f* COMP SUBJ in sentences such as *The guests saw the cake was being decorated*, where a significant garden path effect is not observed. This suggests that neither a retrieval of a different lexical entry for the verb, nor a structural retrieval for reattachment, are taking place. Instead, I propose that the valency of the verb is underspecified in its lexical entry, to allow either OBJ or COMP, and that this in turn affects the constraints on the CAT value of the F-representation of the non-subject daughter. The relationship between the grammatical function and CAT constraints is shown in (184).

(184) a. F MA.GF OBJ $\Leftrightarrow$ F CAT N
b. F MA.GF COMP $\Leftrightarrow$ F CAT V

Modelling this requires some way for the alternation in governed GFs to be specified. This could be in the form of a disjunction or, following Butt (1995), Falk (2005) and others, could describe constraints in terms of feature sets for the composition of GF and the (under)specification of governed GFs. As one option to achieve this, I propose the set of combinations in (185), which is an adaptation from Falk (2005) which was shown in (46) on page 45. The feature ±C is defined as "clausal", and Falk's proposed feature ±S "saturated" is not used.

(185)

		−R	+R
−C	−O	SUBJ	OBL$_\theta$
	+O	OBJ	OBJ$_\theta$
+C		COMP	XCOMP

In this schema, OBJ and COMP are specified [+O, −R, −C] and [+O, −R, +C] respectively. This then allows the OBJ ↔ COMP alternation in the lexical specification of *saw* to be specified as [+O, −R] with no value for C.

In LFG, grammatical functions and lexical categories are generally assumed to be distinct. However, there are some specific lexical items where the relationship between the two is constrained. Examples (186a) and (186b) are taken from Falk (2001, p. 215) and Dalrymple (2017, p. 49, ex. 52) respectively and show category constraints described through f-structure. The F-representation counterparts of these constraints are shown in (187).

(186) a. *help* $[–\text{N}] \in \lambda(\phi^{-1}(\uparrow \text{XCOMP}))$ (note: [–N] = V or P)

b. *become* CAT((↑PREDLINK),%C)
(%C V) = –
(%C P) = –
(%C ADV) = –

(187) a. *help* $(\uparrow \text{XCOMP CAT}) = [\text{ADJ} -, \text{ADV} -, \text{N} -]$

b. *become* $(\uparrow \text{PREDLINK CAT}) = [\text{ADV} -, \text{P} -, \text{V} -]$

The lexical specifications for nouns and verbs then include restrictions on the feature specification of their MA.GF attribute. @N contains the specification MA.GF = [+O, −C], and @V contains the specification MA.GF = [+C], both of which are congruent with the underspecified valency [+O, −R]. To recover the LFG c-structure the template specification for F-representations is amended to reflect the correspondences in (184).

5.3 Other modelling assumptions

The previous sections show how a core incremental theory of LFG can be modelled within the constraints of ACT-R, setting out the assumptions on the structure and content of memory chunks within a model. These assumptions are necessary to represent LFG constraints on incremental structure growth, and to allow an LFG analysis to be retrieved from the output of a model. Developing a model requires other assumptions which are not specific to LFG, but which have an impact on the operation of the model either in terms of the potential for errors, or with regard to the time-course of structure growth.

In this section I set out a number of areas where different modelling assumptions are possible and signal which ones are to be explored in the simulations that are reported in Chapter 6. The areas include the presence and use of additional working memory capacity (Section 5.3.1), managing chunk proliferation (Section 5.3.2), the functions of the Goal buffer in controlling the processing cycle and elsewhere (Section 5.3.3), and the presence of a formal processing stack (Section 5.3.4).

5.3.1 Additional working memory capacity

One question to be addressed is whether additional working memory is necessary in order to model incremental LFG. If there is a case for additional memory, what role should it play in the models?

The models discussed in Chapter 4 all assume additional buffer capacity, which serves a variety of purposes depending on the model, which I return to below. However, the motivation for these buffers was not clear. Jones (2017) reported a model which built functional structure without assuming additional buffer capacity. This was able to build single-embedded sentences but had a capacity constraint such that the time course for relative clauses in English could not be matched qualitatively (i.e. relative size of time course peaks at particular words) nor quantitatively, which suggests that for an LFG model, additional capacity is needed.

The purposes of additional capacity vary between models. Two broad categories of purpose can be identified: creation of new structure, and holding incomplete structure pending further attachment or information. Where new structure is being created, some models assume the presence of specific buffers which creates a specific element of structure. Thus in Lewis and Vasishth (2005), there is a specific buffer for each category/bar level combination required to build structure. This has the advantage that multiple elements of structure can be created simultaneously, and the specialism of a buffer to a node label allows for corresponding productions to be written that are specific to those buffers. This simplifies the task for the modeller, but commits the architecture to a particular set of assumptions regarding c-structure. As discussed in Section 5.2.2.1, this predicts that a reanalysis may be required if the structure initially chosen by the model turns out to be incorrect, with a consequent time cost: such cases are testable empirically. It also reduces the likelihood that the model architecture can successfully parse other languages where the assumed phrase structure rules diverge radically from those for English in the model. From the perspective of the LFG model, these specialisms may relate to the CAT value of an F-representation.

Another role for additional buffer capacity is to hold elements of structure so that they are immediately accessible for processing, rather than requiring a retrieval. This might be the most recently processed word (Van Dyke and Lewis, 2003), or the clause that is currently being built (Chow et al., 2015), or another element of structure where experimental time courses suggest that processing is faster than the time required for an ACT-R retrieval cycle (McElree and Dyer, 2013, p.239).

In Chapter 6, I explore hypotheses around both of these possible roles for additional capacity.

5.3.2 Chunk proliferation

In ACT-R, chunk retrieval does not remove a chunk from declarative memory into the Retrieval buffer. Instead, a copy of the chunk is placed into the Retrieval buffer, with the original chunk remaining in DM. If the copy chunk is released from the Retrieval buffer unaltered, it merges with the original chunk, which gains the activation that has accrued to the copy chunk. However, if the copy chunk is altered in the Retrieval buffer, on release it is not merged with the original, but instead remains in declarative memory alongside the original. The original and the altered copy share many chunk values, and so are potential competitors for future retrieval requests.

This is of particular concern for language processing because of the repeated retrieval and release of chunks that have links added to them. A chunk may have an empty attachment site which is filled in its copy, but the original remains with an empty attachment site. In a noisy system, the original may be copied a second time, with the result that there are now two different attachments in declarative memory. I term this *chunk proliferation.* This is problematic because it seems to generate non-human-like errors: the utterance **I ate the apple the steak* is rejected as ungrammatical, but when the copy of a chunk representing *I ate* has *the apple* added as its object, the original chunk *I ate* is still available to be copied into the retrieval buffer so that *the steak* can be added as its object.

The consequences of chunk proliferation, and possible techniques to reduce unintended outcomes, must be tested through modelling. At the level of overall processing, chunk proliferation during structure-building may not matter if checking for consistency and rejection of parallel structures is carried out at a later stage of processing. It may also be the case that the original chunk may lose activation to such an extent that it cannot be retrieved, in which case potential negative consequences can be ignored.

However, if multiple versions of an F-representation do lead to non-human errors in processing, this must be managed in the model. One possibility for management is to hold the original chunk in a buffer, rather than in declarative memory, until it is released. In this way, there are no competitor copies available. However, a model based on holding incomplete chunks in buffers then raises questions about the role of memory retrieval, which has been used effectively in models such as Lewis and Vasishth (2005). Another possibility is to calculate an explicit candidate set of chunks for retrieval that excludes the original chunks once an amended copy is available. This capability is not part of core ACT-R and would require a list to be maintained and updated alongside the ACT-R production cycle.

A third possibility is to explore the role of spreading activation from the discourse buffer: copy chunks that have more dependents will gain more spreading activation from discourse referents, and this may be enough to suppress the original chunk. However, this requires the referent chunk to be placed into the discourse buffer at the same time that it is attached: if the chunk in the discourse buffer is a different version to the chunk that occupies a structural slot, spreading activation is not available. It might be possible to amend ACT-R so that spreading activation from the discourse buffer accrues not to those chunks that are in the buffer, but to those chunks in declarative memory that share a PRED value with the chunks occupying slots in the discourse buffer. This too is a matter for empirical modelling and will be discussed in Chapter 6.

5.3.3 Controlling the production cycle: the Goal buffer

The incremental theory of LFG sits within a theory of language processing that assumes a processing cycle that applies to each new word that is perceived. Accordingly, an ACT-R model of the theory must therefore reflect this processing cycle and ensure that productions fire in a sequence that is aligned to the theory.

Alongside this requirement for high-level sequencing is the question of lower-level sequencing, which is driven by ACT-R capacity constraints. ACT-R can fire only one production at a time, and the scope of impact of that production is constrained by the overall buffer capacity, representing working memory. Modellers must also bear in mind that, notwithstanding the constraints of buffer capacity, the ACT-R structure may be more powerful than human cognition. Viewed parsimoniously, it is preferable to use smaller productions that undertake one specific element of a processing task, and then model how sequences of productions might be learned, rather than write a complex production at the start. As a result of these constraints, some operations require more than one production to fire in sequence. This is distinct from production compilation, which builds chains of productions that often follow each other. At these points, there is a risk of unintended action from the model, if an unrelated production fires at a point when the intended production sequence is not yet complete.

One option to control the sequence of production firing — at both high and low levels — is to use a dedicated slot in the Goal buffer, the presence of which enables some productions to be fired and inhibits other productions from firing. However, that itself may not be a cognitively plausible assumption as it raises the question about how such processes are learned. Alternatively, productions may be written in such a way that the conditions for a production to fire (in buffers other than the Goal buffer) mean that some chunk sequences are impossible. The effect of both of these strategies can be tested empirically, and is explored in Chapter 6.

5.3.3.1 Other functions of the Goal buffer

The Goal buffer can also be used to hold other flags, such as the presence of an open long-distance dependency and perhaps high-level information about the nature of the anchor, such as category or grammatical function (e.g. Wagers, 2013; Wagers and Phillips, 2014).

If there is no dedicated buffer capacity, the Goal buffer may also hold chunks, or references to chunks, for which rapid access is required during processing, as discussed in Section 5.3.1.

5.3.4 Processing without an explicit stack

Parsing algorithms frequently make use of stacks, and stacks are present in some of the models described in Chapter 4, e.g. Hale (2014); Stewart et al. (2014). There is no formal stack structure within ACT-R and so where a stack is part of an algorithm, a solution within ACT-R must be found. One solution is to use a stack variable alongside the memory chunk structure (e.g. Anderson et al., 2001) or to use a slot in the Goal buffer to store either a chunk which is awaiting processing, or to store an expectation of future content, which can be compared with model input, and which is changed by productions depending on the processing path followed by the model (e.g. Lewis and Vasishth, 2005).

The incremental LFG model described in Chapter 6 does not make use of a stack, but also assumes that once a chunk has been released to declarative memory, it is no longer retrievable for further attachment. Rather than a stack, the model explores the use of ACT-R's multibuffer capability to maintain multiple active chunks in working memory. The contribution and drawbacks of this approach will be discussed in more detail in Chapter 6.

Chapter 6

Implementing an LFG-based model in ACT-R

This chapter applies the principles that were developed in Chapter 5 to develop and test an ACT-R computational model of the incremental processing algorithm developed in Chapter 3. In this chapter I aim to demonstrate that the model faithfully adheres to the incremental theory set out in earlier chapters and that it can thus be used to test the predictions of the theory about interactions between previously-processed lexical information and the constraints introduced by subsequently-encountered words.

ACT-R is often used to model cognitive time-courses, but the focus of this project is more on the content of structural representations and their relationship to a theory of grammar. Testing the model addresses two questions.

1. To what extent does the model represent the incremental theory, in that the model's representation of the grammatical constraints and structure-building process proposed in Chapters 3 and 5 generate representations that match LFG analyses, and that the process is valid in typologically different languages?

2. To what extent does the model replicate the predictions of the incremental theory that were identified in Chapter 3?

The operation of the model is constrained by the design of ACT-R: results of the testing illustrate the impact of those constraints. Those limitations will be discussed at the end of the chapter.

The chapter is in four parts, which cover the scope of the model, the structure and function of declarative memory, and the procedural system. In the first part, Section 6.1, I set out the architectural specification of the model, including the assumed buffer capacity, and the nature of memory representations of lexical information and of syntactic structure. In doing so, I show how these relate to the incremental theory, and how the lexical memory chunks are derived from LFG lexical specifications together with phrase-structure rules for the grammar. This includes feature-based representations of category and grammatical functions, which allow argument alternations to be processed without requiring reanalysis. Examples are given for words from different categories in English and Korean, with the full model vocabularies available in the electronic supporting documentation[1]. I also show how the specification of structural chunks relates to the incremental theory as well as satisfying the functional requirements of ACT-R.

In the second part, Section 6.2, I describe the operation of the model in detail, including the processing cycle, and the production set that manipulates memory chunks. I also detail the additional Lisp functions that support the operation of the model. Within this part of the chapter, the relationship between the operations derived in Chapter 3, the algorithm set out in Chapter 4 and the production set is made clear. The full code for the model implementation is available as part of the above-mentioned supporting documentation.

The third part of the chapter presents the results of testing the model on a variety of sentence types from English and Korean. In Section 6.3, I show how the model represents simple and complex syntax, before moving on in Section 6.4 to model specific predictions of the incremental theory relating to context-dependent processing, the impact of the LFG principles of completeness, coherence and consistency, and the role of prosody in supporting the processing of complex structure.

Finally, in Section 6.5, I discuss the constraints and limitations of the model and relate these to the modelling assumptions that I have made, and to the architectural assumptions that are inherent in ACT-R.

6.1 Specification of the model

The aim of the model is to represent a hypothesis about a structured interface between lexical information and other aspects of cognition, and about the calculations that take place to generate content using this interface bottom-up during incremental sentence processing. The model is based on the modular projection architecture of Lexical Functional Grammar and is language-independent, in that it tests sentence processes in two typologically different languages, English and Korean.

The model does not represent the whole of the grammar of English or Korean, but uses a subset of the grammars of the two languages that is sufficient to process the test items set out in Section 6.3. Specific computations that the model should carry out include:

1. analyse simple sentences, including attachment of subject, object, and prepositional object where selected by the verb;
2. analyse relative clauses modifying a noun, with or without an explicit relative pronoun, where the grammatical function of the pronoun within the relative clause may be subject, object, or prepositional object (King and Just, 1991);

---

[1] `https://deposit.ora.ox.ac.uk/datasets/uuid:79cc0e91-1f68-4772-88e9-0e3e7e9c5312`

3. analyse subordinate clauses with an explicit complementiser, where the COMP grammatical function is selected by the verb.

6.1.1 Input, output, measurements

Data input is modelled using the ACT-R Visual module. Thus, the model is structured such that it could be aligned to an eye-tracking paradigm, although it is not seeking to model eye-movements during processing.

Two metrics are generated by the model. One is a record of the structure that is generated by the model after each word has been processed. The second is the model cycle time for each word, which is in turn defined as the time that it takes to model the processing of each word, from the point that visual attention focuses on one word until visual attention focuses on the next word.

Outputs for each run of the model are stored in a specific, timestamped directory, and consist of:

- a .csv file containing the cycle time data;
- a .gv file for each cycle plus one for the final output that contains data on the structural chunks and their relationship in a format that allows a GraphViz (Bank, 2013) visual representation to be generated.

The code includes a script for converting each .gv file in the directory to a graphical .png file using the *dot* language (Gansner et al., 2015).

6.1.2 Building lexical memory

Declarative memory in ACT-R was discussed in Chapter 5. The following conventions are used in the text that follows.

- the names of buffers have an initial capital, or are in all capitals if the name is an abbreviation rather than a word, e.g. Goal, Retrieval, VP;
- the names of chunks and of slots within chunks are in fixed-width font, lower case, e.g. `dep1.gf subj ma chunk0-0`
- the logical values True and False are `t and nil` respectively
- Lisp string objects are in italics, e.g. *senator*
- the names of productions are shown in fixed-width capitalised, e.g. `CREATE-VERB-FOR-NV`.

In the model, lexical memory represents language-specific knowledge of words and their combinatorial constraints. Each word is represented as a single memory chunk of type `lex-ent` which is present in declarative memory at the start of a run. Lexical memory chunks are not created during processing. The slots in a lexical chunk can be divided into four groups: semantic and category information about the word itself; category and grammatical function constraints that apply to the attachment to its mother pre- or post-lexical head; grammatical function specification of required and optional daughters, including any argument alternations; and ordering constraints on the GF of dependents in relation to the head. To illustrate these groups, the lexical specifications for English *insulted* and Korean *moyokhayssta* 'insulted.PLAIN' are given in Table 6.1[2].

Table 6.1: Lexical chunk specifications in English and Korean

1.	`insulted0`		`moyokhayssta0`	
2.	`isa`	`lex-ent`	`isa`	`lex-ent`
3.	`string`	*insulted*	`string`	*moyokhayssta*
4.	`predreqd`	`no`	`predreqd`	`no`
5.	`pred`	*insult*	`pred`	*insult*
6.	`cat`	`((v plus) (n minus)`	`cat`	`((v plus) (n minus)`
7.		`(adjcv minus)`		`(adjcv minus)`
8.		`(adv minus)`		`(adv minus)`
9.		`(p minus))`		`(p minus))`
10.	`ma`	`t`	`ma`	`no`
11.	`ma.cat`	`((adjcv minus)`	`ma.cat`	`(nil)`
12.		`(adv minus)`		
13.		`(p minus))`		
14.	`pre.ma.gf`	`((r minus))`	`pre.ma.gf`	`no`
15.	`post.ma.gf`	`((c plus))`	`post.ma.gf`	`no`
16.	`subj`	`reqd`	`subj`	`orpro`
17.	`obj`	`reqd`	`obj`	`orpro`
18.	`obl`	`no`	`obl`	`no`
19.	`obj-th`	`no`	`obj-th`	`no`
20.	`comp`	`no`	`comp`	`no`
21.	`comp-poss`	`t`	`comp-poss`	`no`
22.	`valency`	`2`	`valency`	`2`
23.	`ggf1`	`((o minus) (r minus)`	`ggf1`	`((o minus) (r minus)`
24.		`(c minus) (g plus))`		`(c minus) (g plus))`
25.	`ggf2`	`((o plus) (r minus)`	`ggf2`	`((o plus) (r minus)`
26.		`(c minus) (g plus))`		`(c minus) (g plus))`
27.	`ggf3`	`no`	`ggf3`	`no`
28.	`ggf1.stat`	`reqd`	`ggf1.stat`	`orpro`
29.	`ggf2.stat`	`reqd`	`ggf2.stat`	`orpro`
30.	`ggf3.stat`	`no`	`ggf3.stat`	`no`
31.	`pcase`	`no`	`pcase`	`no`
32.	`adj`	`poss`	`adj`	`poss`
33.	`gf.prec`	`ggf1`	`gf.prec`	`no`
34.	`gf.post1`	`ggf2`	`gf.post1`	`no`
35.	`gf.post2`	`no`	`gf.post2`	`no`
36.	`adj.post`	`t`	`adj.post`	`no`

The first group (rows 2–9) contains the semantic information, as well as specifying the category of the structural chunk to which this information can be added. The `predreqd` slot has value `t` for functional categories such as determiners.

The second group of slots (rows 10–15) shows the constraints on the f-representations to which the word can attach. These are derived from the LFG preceding and succeeding

---

[2] The line numbers in Table 6.1 are for reference and have no function in ACT-R.

mother sets that were introduced in Chapter 3. The MA.CAT specification for *insulted* represents an underspecified feature set. The given values are `((adjcv minus) (adv minus) (p minus))`: no values are given for the features `n` and `v`. As a result, an f-representation created from this lexical specification can can be a daughter of an f-representation where either value of the `n` or `v` feature is `plus`. The `pre.ma.gf` and `post.ma.gf` show the constraints on the grammatical function that the word can provide. If it attaches to a structural chunk before that chunk's PRED value is provided, it is congruent with a grammatical function whose feature specification includes [R–], whereas if it attaches after the PRED value is provided, it is congruent with grammatical functions whose feature specification includes [C+]. In LFG terms, using the grammatical function feature specifications in Section 5.2.7, this permits *insulted* to provide the pre-head SUBJ, OBJ, COMP or ADJ, or the post-head COMP, XCOMP or ADJ for a verb or noun[3]. Successful attachment depends on the grammatical function constraints for the daughter also being congruent, and these further restrict attachment possibilities. For *moyokhayssta*, which according to the grammar in Appendix A has TYPE:NO, attachment as the daughter of another chunk is never possible.

The third group of slots (rows 16–32) sets out the grammatical function constraints on daughter chunks. Analogously to the slots in the second group, these are derived from the preceding and succeeding daughter sets introduced in Chapter 3. The information is provided in two ways, each of which serves different purposes in the model. The first way itemises the governable grammatical functions considered in the model (SUBJ, OBJ, OBL, OBJ$_\theta$, COMP) and specifies the extent to which they are required in order for the principles of completeness, coherence and consistency to be satisfied. Thus a value of `reqd` indicates that a grammatical function is required, whereas `no` indicates that it cannot be present. The value `orpro` models argument drop: if there is no dependent providing a grammatical function, a PRO dependent is inserted by the model. The value `alt` indicates that an alternation is available between the grammatical functions carrying that value.

Governed grammatical functions are also specified in the series of `ggf` slots, whose value is either `no` or a feature specification. In the examples in Table 6.1, the governed grammatical functions are all fully specified, but underspecification is also possible if there is an alternation. For example, in the model's lexical specification for *gave*, which corresponds to the LFG specification 'gave ⟨SUBJ, OBJ, {OBJ$_\theta$ | OBL}⟩', the value of slot `ggf3` is `((r plus) (c minus) (g plus))`, underspecified for the feature [O]. This allows for the function to be provided by either a chunk of category P with `post.ma.cat` specified as `((o minus) (r plus) (c minus))`, or a chunk of category N with `post.ma.cat` specified as `((o plus) (c minus) (g plus))`, but not both. The `ggf.stat` slots show the status of the governed function with respect to completeness, coherence, and consistency. Finally, the slots `pcase` and `adj` hold any specification relating to the nature of oblique arguments,

---

[3] The grammar rules in Appendix A allow a preposition to select a COMP argument, but this is not included in the model.

and whether adjunction is allowed.

The fourth group of slots (rows 33–36) carries ordering information, setting out which if any of the word's governed grammatical functions is constrained to appear immediately before, or immediately after the head.

Memory chunks are used in the model in only one phase of the processing cycle. More details on how this operates are given in Section 6.2.

6.1.3 Memory chunks for sentence structure

Emerging sentence structure is represented by memory chunks of type `f-rep` that correspond to the f-representations introduced in Chapter 5. In addition to the information about combinatorial constraints, which is transferred to structural chunks from lexical specifications, the chunks hold information about the dependents of the f-rep and the grammatical functions that they provide, in the `dep` and `dep.gf` series of slots. The `dep.gf` slots represent an ordered sequence and hold any grammatical function ordering constraints that apply. The slots are not intrinsically ordered within the chunk: specific constraints in the production set serve to ensure that dependents are added in the intended order (e.g. if `dep1` is unfilled, add a value to `dep1`; if `dep1`, `dep2`, and `dep3` are filled, add a value to `dep4`).

Structural chunks also have the slots `df`, which holds any chunk that has a discourse function, and the slot *embedded*, which is has the value `t` when the value of the `pred` slot is the head of an embedded clause.

6.1.4 Working memory capacity and role

The theory that the model is testing is a theory of structure building rather than a theory of a particular memory search strategy. The memory structures within the model are deliberately not restricted to binary branching, reflecting functional structure in LFG, and information may be added to a structural chunk over the course of many words. Attempts within this project to realise a model using retrievals from declarative memory, similarly to the approach of Lewis and Vasishth (2005), were unsuccessful. This is in large part because a chunk to which new information has been added co-exists in declarative memory with the previous version of the chunk which does not have the new information. The consequent proliferation of competitor chunks significantly reduced the accuracy of the structural representations generated by the model. Resolving the issues arising is outside the scope of this thesis: there is more detailed discussion of the issues and potential avenues for resolving them in Section 6.5.

6.1.4.1 Buffers to build structure

Instead of retrievals, structural chunks are built and maintained in buffers until they are closed because no further information can be added. Previous work (Jones, 2017) showed that it was not possible to build LFG functional structure in ACT-R using only the core buffer infrastructure, and accordingly the model makes use of three additional, non-core buffers, named AP, NP, and VP. The NP and VP buffers are used principally for f-representations of nouns and verbs, respectively, with the AP buffer being used for other word classes. These buffers are based on the Goal-style buffer template that is available as part of ACT-R. Goal-style buffers allow the creation of new chunks, possibly with an associated time cost, and for the slots in a chunk to be amended while the chunk is within the buffer. However, the additional buffers cannot make a retrieval from declarative memory.

The VP buffer creates chunks with the feature `cat v plus`, the NP buffer with the feature `cat n plus` and the AP buffer with the `cat` feature `p plus`, `adjcv plus`, or `adv plus`.

A chunk is created in one of these buffers in one of three circumstances:

(i) when a lexical memory chunk is retrieved with a word of that category;

(ii) when a lexical memory chunk is retrieved with a functional category (D, I, C) that is constrained to appear with one of the lexical categories, and there is no chunk in the buffer to which the information from the functional memory chunk can be added;

(iii) when another structural chunk is ready to attach and there is no chunk in the buffer corresponding to its `ma.cat` value to which the attachment can be made.

As soon as a `ma.cat` value has been added to a structural chunk, that chunk must attached to another, unless that value is `ma.cat no`. Chunks may also be attached without a `ma.cat` value if there is a potential mother chunk already present that has an empty dependent slot whose grammatical function is compatible with the constraints on the daughter chunk. If attachment is not possible, the chunk remains in the buffer unattached.

Structural chunks are released from the buffers into declarative memory when they are attached and have a semantic index (i.e. `predreqd nil`), and either

(i) it is not possible to attach further chunks after the PRED (i.e. the chunk carries the following slot-value pairs: `gf.post1 no` / `gf.post2 no` / `adj.post no`); or

(ii) it is possible to add further information but the following word is constrained such that it cannot be attached into the chunk.

6.1.4.2 The VP buffer as a multibuffer

Clausal embedding, which in the model is represented by a `cat verb` chunk attached to another structural chunk, is widespread and often requires the model to maintain two incomplete structural chunks at the same time. Chunks with `cat verb` may attach to chunks with `cat noun` to represent relative clauses, or to chunks with `cat verb` to represent subordinate clausal arguments or adjuncts of a matrix clause. A different solution is used to represent this clausal embedding: the VP buffer is specified as a multibuffer.

A buffer specified as a multibuffer is able to maintain more than one chunk in working memory as part of a multibuffer set, where this is needed for a specific processing task (e.g. the "threaded cognition" model of multitasking, Salvucci and Taatgen, 2008), rather than a chunk being released into declarative memory for later retrieval. Only one of the chunks in a multibuffer set can be active at a given time, but other chunks in a set can have their slot values changed without proliferation of copy chunks in declarative memory. To be part of a multibuffer set, a chunk must never have been released to declarative memory. It is possible to search the multibuffer set so that one of the chunks is in focus, but the search mechanism is not exhaustive: in other words, once a chunk has been identified within the set that matches the search cues, it becomes the active chunk in the buffer. This is different to the search mechanism for the Retrieval buffer, where all matching chunks are identified and the chunk with the highest activation is selected.

Although the VP multibuffer allows several incomplete structural chunks to be maintained in working memory, it is not the same as a stack: there is no ordering of chunks within the multibuffer set and so no guarantee that the last chunk added to the set will be the next chunk brought into focus as the active chunk, which would be a characteristic of a stack. Thus as a multibuffer set increases in size, the likelihood increases that it will behave differently to a stack. There is also no inbuilt time penalty within a multibuffer for bringing a different chunk into focus. For a retrieval from declarative memory, the number of competitor chunks and the level of activation of those chunks both affect the time course of the retrieval. This is not the case for a multibuffer. This model is not seeking to replicate processing time courses and so the timing impact of multibuffers is moot. However, it is possible in future research to model different ways in which multibuffer set membership affects the time course and test these hypotheses empirically.

6.1.4.3 Controlling the processing cycle: the Goal buffer

The ACT-R Goal buffer can be used to hold information about the desired outcome of a processing task, to steer the processing pathway by constraining productions, and to hold intermediate outputs or information that will be accessed later in the task. A chunk is placed in the buffer, with the values of particular slots changing over time. In the model, the last two of these functions are used.

Figure 6.1 shows the processing cycle adopted by the model, with the values for the goal-state slot that are used in each stage.

In addition to tracking the goal-state, the Goal buffer keeps a record of the locus of structure-building, also referred to as the buffer in focus, which is either the last buffer to which information was added, or if that buffer has been cleared, the buffer holding

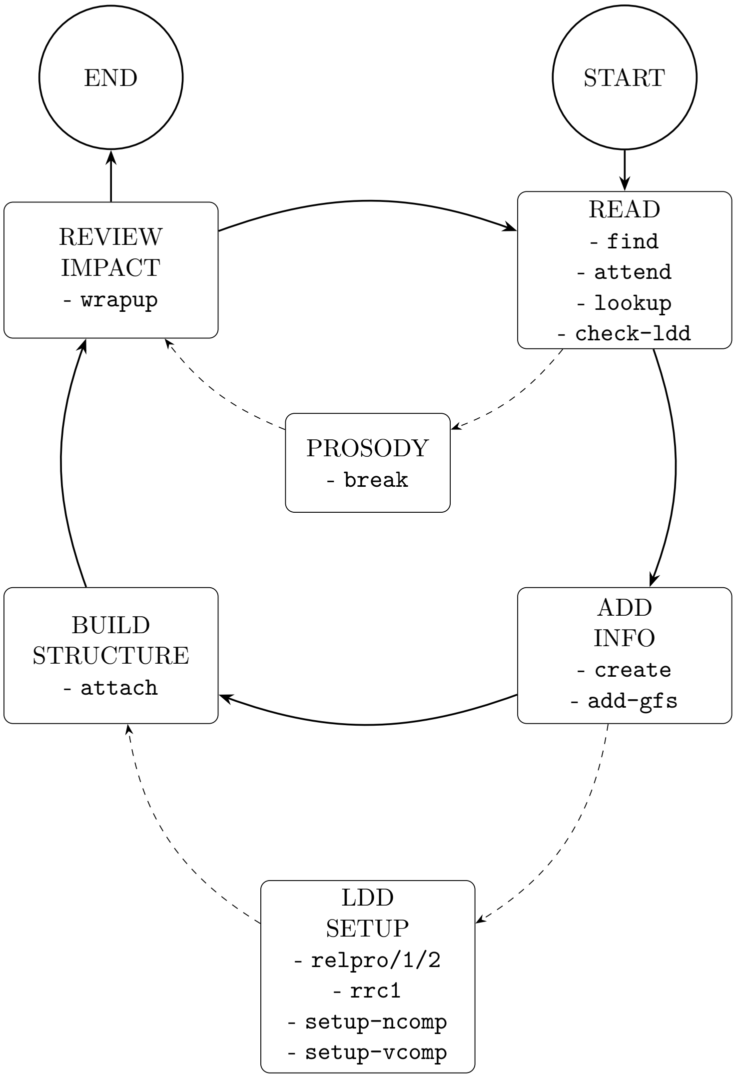


Figure 6.1: The model cycle with goal states

the mother of the chunk that was cleared from the buffer. Further slots in the Goal buffer, `lddopen, lddtype, ldd, holdid1, holdid2, holdid3, holdid4` are used for when building complex structure and will be discussed in more detail in Section 6.2.1.

6.2 Operation of the model

Within the architecture set out in Section 6.1 there is still considerable freedom of choice in the details of how memory chunks are retrieved and manipulated to build structure. This section sets out the operation of the model, including the processing cycle and its link to the algorithm proposed in the incremental theory. It relates the model's production set to this cycle and discusses how particular subsets of productions operate. In addition to ACT-R's pattern-matching capacity, the productions make use of Lisp functions, and the role of these is also described.

6.2.1 Implementing the algorithm

The mapping between the incremental parsing algorithm and the model algorithm can be summarised as follows. The core parsing cycle is steps 1–4. This is language-independent and the same steps are followed for English and Korean.

1. READ

   (i) Read next word from screen (goal states `find, attend`) and access lexical memory chunk (goal state `lookup`).

   (ii) If an LDD has been provisionally closed, check this against the next word's category and confirm or undo the provisional closure (goal state `check-ldd`).

2. ADD INFO

   (i) If possible, add lexical information directly into the chunk in focus. If the relevant buffer for the word category is empty, create a new chunk in that buffer and add the information to it.

   (ii) If the lexical information cannot be immediately added to a chunk in a buffer, compare the category constraints from the lexical memory chunk with the category information on the structural chunk in focus, determine whether the new information is to be added within the structural chunk in focus, or whether focus shifts outside.

   (iii) If focus shifts such that the current chunk or one of its daughters is no longer in focus, find an appropriate chunk to be the chunk in focus, wrap-up the current chunk, and repeat the previous step.

   (iv) If focus does not move outside the current chunk, create a new chunk daughter within the current chunk. Trigger LDD SETUP goal states if necessary.

   (v) Add grammatical function constraints on dependents (goal state `add-gfs`).

3. BUILD STRUCTURE

   (i) Find or create an attachment site if one is not already present in another buffer (including cases where attachment to a chunk present in a buffer would result in an incoherent structure).

   (ii) Attach the chunk and assign grammatical function if possible (goal state `attach`).

4. REVIEW IMPACT

   (i) Complete housekeeping on assigning grammatical functions, and provisionally close a long-distance dependency if possible (goal state `wrapup`).

(ii) If chunk in focus is closed, clear it from the buffer and move focus to the buffer holding its parent. Return to the start of the cycle.

5. PROSODY

   The goal state `break` is triggered if a representation of a prosodic break '||' is read from screen.

6. LDD SETUP

   The restricted buffer capacity of the model means that some operations creating embedded clauses require more than one production to fire in sequence, with attachment information being temporarily held in slots in the Goal buffer. These are managed through goal states `relpro/relpro1/relpro2` for relative clauses with a complementiser, `rrc1` for restricted relative clauses, and `setup-ncomp` and `setup-vcomp` for COMP dependents of nouns and verbs respectively.

The elements of the algorithm are discussed in more detail below. The full model code is available in the electronic supporting documentation[4].

6.2.1.1 The READ goal states

This element of the algorithm is governed by `goal-state` values `find, attend, lookup`.

A word is read using the ACT-R visual module in two stages. ACT-R maintains a representation of objects in the visual field, with a single locus of visual attention determining which object in the visual field can be attended. The production `FIND-NEXT-WORD` moves the locus of visual attention to the next word to the right, or to the leftmost word in the next line if there is no word to the right. The production `ATTEND-WORD` generates a memory representation of the word that is at the current locus of attention. The default ACT-R specification of the time to generate a representation is 85ms and the model does not assume anything different.

Once a memory chunk representation is available, one of the `GET-LEX-ENT` productions fires, depending on the value of the `lddopen` slot in the Goal buffer. The productions search for a `lex-ent` chunk whose `string` slot matches the `string` slot of the chunk in the Visual buffer, and delivers the best match to the Retrieval buffer. If an LDD has previously been provisionally closed, the Goal state moves to `check-ldd`, where the `UNDO-PROV-LDD` and `CONFIRM-PROV-LDD` family of productions assess whether the next word displaces a provisionally-assigned LDD. Once this is done, or if there was no provisional LDD assigned, processing moves to the ADD INFO stages.

Reading a prosodic break || from screen causes the production `PROSODIC-BREAK` to fire, moving the parser into the `break` goal state. Reading an asterisk $*$ triggers the `TEXT-END-WRAPUP` production which processes any remaining content in the buffers and the multibuffer set before the run ends.

---

[4] `https://deposit.ora.ox.ac.uk/datasets/uuid:79cc0e91-1f68-4772-88e9-0e3e7e9c5312`

6.2.1.2 The ADD INFO goal states

Productions in the `create` goal state do the bulk of the work of assessing the fit between the retrieved lexical item and the existing structure, in terms of the category and grammatical function specifications on the relevant chunks.

The `NEW-FREP` and `ADD` families of productions fire if the relevant structural buffer is empty, or if the chunk in the buffer has capacity to hold information from the retrieved lexical entry. This latter case might happen if a `cat (v plus)` chunk has been created as the mother of another chunk but does not yet have a value for its PRED slot, or if a determiner has created a `cat (n plus)` chunk that needs a `pred` value from a noun. If the VP buffer is empty but the multibuffer set holds a chunk, this is moved into the VP buffer with the `FILL-VP-FOR-VERB` production.

In some cases, the retrieved lexical entry signals that the chunk in focus is closed, with focus then shifting to that chunk's parent. The `VERB-CLOSES` and `VP-NEEDS-PREP` families of productions and `CLEAR-NP-FOR-NP-AS-DEP3` do this.

If embedding is necessary, the production families `START-EMBEDDED-COMP`, `CREATE-COMP`, `RELPRO-SETUP`, `CREATE-RELPRO-CHUNK`, and `NP-AFTER-NP-TRIGGERS-RRC` move to the appropriate LDD SETUP goal states.

Once the lexical information has been added, productions in the `add-gfs` goal state fire. These productions, from the families `GF-PREC`, `AP-GF`, `ADD-GF-POST`, and `VP-POST-GFS-ALL-ADDED` families transfer any lexically specified ordering constraints on daughter grammatical functions from the `gf.prec`, `gf.post1`, and `gf.post2` slots in the lexical entry to the `depX.gf` slots in the structural chunk. Families of parallel productions are needed because ACT-R slots are not ordered within a chunk, and so it is not possible to write a single abstract production that covers the different possibilities for adding information. These productions also close a long-distance dependency if a relative pronoun is immediately followed by a verb (i.e. a subject relative clause in English).

6.2.1.3 The BUILD STRUCTURE goal states

These productions are governed by the goal state `attach`. Productions within the production families `ATTACH-NP-TO-VP, ATTACH NP-TO-AP, ATTACH-VP-TO-NP, ATTACH-AP-TO-VP AND ATTACH-AP-TO-NP` compare feature sets between chunks in the relevant buffers and create attachments based on those constraints. (The AP buffer is used for prepositions, adverbs and adjectives including Korean adjectival verbs). Where a buffer is empty but attachment is required, the buffer is filled by creating a new empty chunk. Potentially incoherent structures arising from multiple dependents with the same feature specification are identified at this point.

6.2.2 The REVIEW IMPACT goal states

Once a chunk has been attached, the productions in goal state `wrapup` fire, assessing what action needs to be taken on the chunk in focus. If the chunk is still open, in that it requires dependents or a `pred` value, or if adjunction is still possible, no action is taken and processing continues. If a LDD can be provisionally assigned, this takes place. If a chunk is closed, either by its own lexical specification or because a word has been read that closes the chunk, completeness and coherence are assessed. This takes place by comparing the feature specification of attached dependents with the selected grammatical functions and assigning a fully-specified grammatical function if that is not already there.

Once this is complete, the parsing cycle starts again with the READ productions.

6.2.2.1 The PROSODY goal states

The goal state `break` is reached only if the symbol ||, representing a prosodic break, is read from screen. Productions in this goal state are are from the family `PROSODIC-CLEAR`. These act to shift focus: if the current chunk in focus has a PRED value, it is closed and cleared from the buffer, with focus moving to its parent. Wrap-up takes place at this point if necessary. If the chunk in focus does not yet have a PRED value, embedding is triggered: focus moves to a new clause that is the daughter of the previous chunk. Once focus has changed, the next word is read.

6.2.2.2 The LDD SETUP goal states

These productions create and attach new chunks required for embedding, whether as relative clauses with a complementiser (the `RELCLAUSE-ATTACH` and `NOTE-RELPRO-CAT` productions), reduced relative clauses (the `RRC-ATTACH` productions) or as COMP dependents of nouns or verbs (`COMPLETE-COMP-OF-NOUN`, `COMPLETE-EMBEDDING-COMP-OF-VERB`).

6.2.2.3 Technical productions

The production `INITIALISE` fires only with goal state `start` at the beginning of a run. The production `END-PARSE` ensures that all buffers are clear, closes the ouptut stream to the .csv file that holds time course information, and generates a graphic representation of the final parse.

6.2.3 Lisp functions used in the production set

Values for categories and grammatical functions in the model are composed of atomic feature sets. Because the combinatory constraints on mothers and daughters are often expressed using underspecification, there are many potential combinations of feature value to be tested. Implementing this using ACT-R productions would require many parallel productions in a family, and to avoid this, Lisp functions have been written to assess fit between category sets. These include the tests `congruentp`, `list-slot-congruentp`, `attachablep` and `not-neg`, which compare either whole feature sets or specific feature values. Lisp functions `nonuniquep` checks whether a potential dependent would create an incoherent representation with two identically specified dependents. Function `prodrop` reduces the need for multiple parallel productions within the `wrapup` goal state, when assigning chunks representing pronouns to unfilled grammatical functions where pro-drop is specified as an option.

The function `compilable-empty` mimics the ACT-R query `buffer empty`, which is used as a LHS condition in productions, but which prevents a production being compiled when production learning is switched on.

There are also functions that relate to the VP buffer as a multibuffer. The function `buffer-set-empty` returns `t` if its argument is a multibuffer with no chunks in its buffer set, and `nil` otherwise. It is used in a number of productions to ensure that a new chunk is not created in VP if there is a chunk in the multibuffer set that might act as a host for new information.

The pair of functions `one-chunk-in-buffer-set` and `fill-buffer-from-set` are used in the production `FILL-EMPTY-VP-FROM-SET`.

The function `clear-multibuffer-to-dm` is not called by any productions during structure-building, but instead is used at the end of the model run: it ensures that all chunks created in VP are released into declarative memory so that queries using the ACT-R environment interface are based on complete information.

Lisp functions also support data collection and generation of the model output. The function `add-to-chunk-set` maintains a list of structural chunks created by the model. This list is used as the input to function `generate-graph`, which returns a Graphviz file with a visual representation of the structure. Functions `interim-graph` and `new-output-graphfile-with-index` produce files with these graphs during the sentence and on completion respectively. Function `update-message-csv1` outputs information on the time course to a .csv file for statistical analysis.

The code for all functions is available in the electronic supporting documentation[5].

6.3 Testing the model: building structure

The model generates a representation of an utterance after each word, and these intermediate representations are stored in output files. The output format from the model differs visually from the usual depiction of f-structure as an attribute-value matrix, although both f-structure and the model's output are directed graphs. Each node in the graph is generated by a memory chunk that holds a semantic PRED value. The text in the node is the chunk name and the PRED value. Chunk names are generated automatically by ACT-R and the number shows the order in which chunks were created: this has no impact

---

[5] `https://deposit.ora.ox.ac.uk/datasets/uuid:79cc0e91-1f68-4772-88e9-0e3e7e9c5312`

on the meaning representation, but could be used to reconstruct a full c-structure, as it allows memory chunks to be related to ordered phrase-structure rules. The topmost node of the graph corresponds to the outermost f-structure. Labels on the arcs between nodes correspond to the grammatical function attributes of f-structure.

Figure 6.2 shows the intermediate representations generated during the processing of the sentence *The senator who the error embarrassed attacked the reporter*, discussed further at example (197) on p. 191. During processing, a grammatical function may be partially specified; where this is the case, the arc labels show the available feature values.

The test items in this section and Section 6.4 have been selected to address the questions posed at the start of the chapter about the functioning of the model and the predictions of the incremental theory. Each item is introduced with the predicted model output alongside the actual model output, together with a discussion of any particular characteristics that have an impact on how the model functions. The predictions are presented as f-structures: these analyses use the set of phrase structure rules and lexical entries in Appendix A.

6.3.1 Simple sentences in English and Korean

The following sentences test the core functioning of the model, including the ability to process argument alternations by combining feature specifications. I begin by presenting the English sentences.

Unless otherwise noted, the results presented below show the output at the end of the sentence. Intermediate outputs are available: once a chunk is attached and/or has a PRED value, it is added to the structure.

6.3.1.1 Simple sentences in English

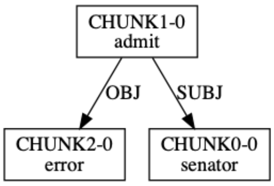
Sentence (188) is a simple transitive sentence.

(188) The senator admitted the error.

Prediction

$$\begin{bmatrix} \textsc{pred} & \text{`admit}\left\langle \textsc{subj},\ \textsc{obj}\right\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{`senator'}\end{bmatrix} \\ \textsc{obj} & \begin{bmatrix}\textsc{pred} & \text{`error'}\end{bmatrix}\end{bmatrix}$$

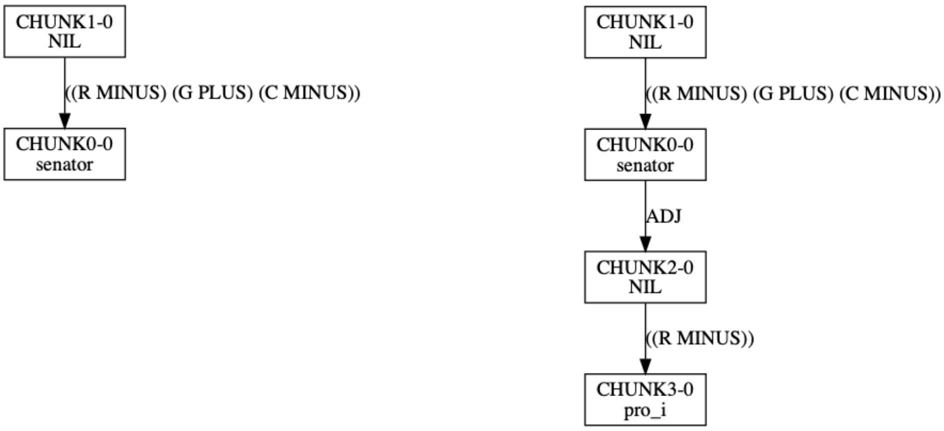
Model output



The verb *hope* in sentence (189) selects SUBJ and obligatory OBL grammatical functions.

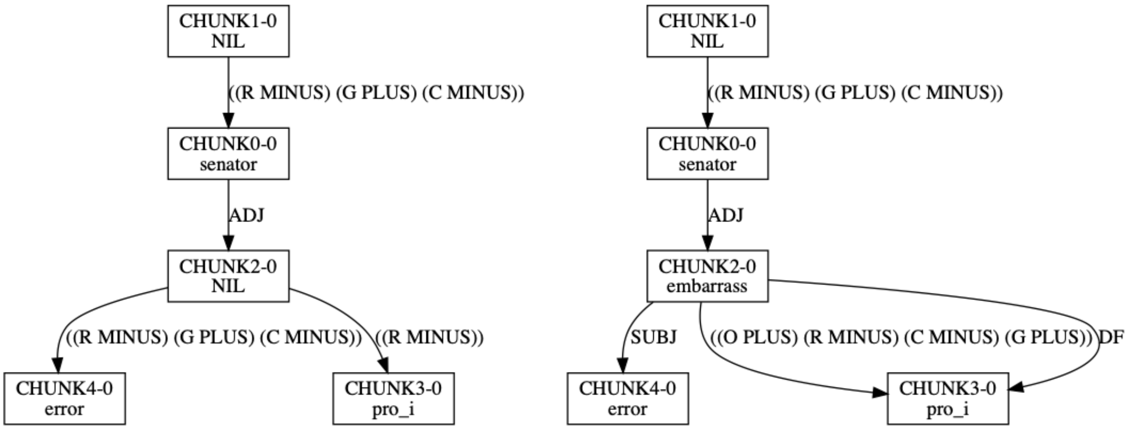
The senator...

The senator who...



The senator who the error...

The senator who the error embarrassed...



The senator who the error embarrassed attacked...

The senator who the error embarrassed attacked the reporter.

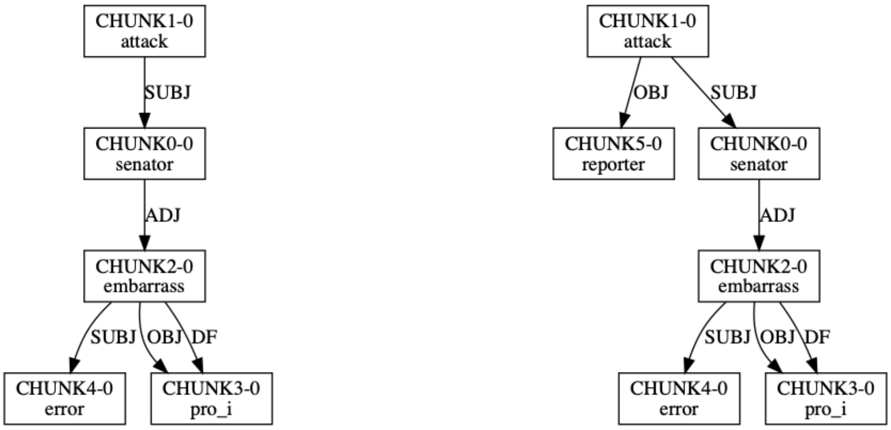


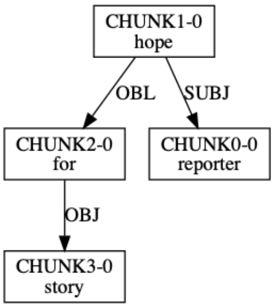
Figure 6.2: Intermediate representations generated during processing

(189) The reporter hoped for a story.

Prediction

$$\begin{bmatrix} \textsc{pred} & \text{`hope}\left\langle \textsc{subj},\ \textsc{obl} \right\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{`reporter'} \end{bmatrix} \\ \textsc{obl} & \begin{bmatrix} \textsc{pred} & \text{`for}\left\langle \textsc{obj} \right\rangle\text{'} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`story'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Model output



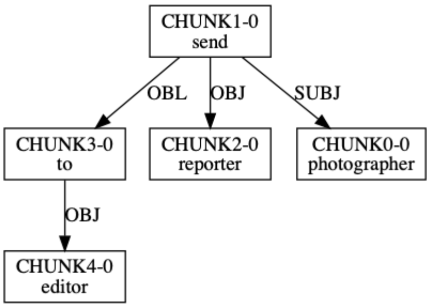
In sentence (190), *send* selects SUBJ, OBJ and OBL grammatical functions.

(190) The photographer sent the reporter to the editor.

Prediction

$$\begin{bmatrix} \textsc{pred} & \text{`send}\left\langle \textsc{subj},\ \textsc{obj},\ \textsc{obl} \right\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{`photographer'} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`reporter'} \end{bmatrix} \\ \textsc{obl} & \begin{bmatrix} \textsc{pred} & \text{`to}\left\langle \textsc{obj} \right\rangle\text{'} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`editor'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Model output



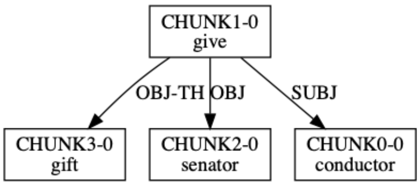
The next two test items explore the OBL/OBJ$_\theta$ alternation for the third argument of *give*. Sentences (191) and (192) present the two variants: the prediction is that a single lexical specification for *give* can generate the two variants, and so other than the lexical content of the OBJ argument, the processing pathway should be identical until the category of the third argument is known. The alternation is reflected in different sets of grammatical functions, and in different mappings from f-structure to s-structure argument position.

(191) The conductor gave the senator a gift.

Prediction

$$\left[\begin{array}{ll} \textsc{pred} & \text{`give}\left\langle \textsc{subj},\ \textsc{obj},\ \left\{\textsc{obj}_\theta \mid \textsc{obl}\right\}\right\rangle\text{'} \\ \textsc{subj} & \left[\begin{array}{ll}\textsc{pred} & \text{`conductor'}\end{array}\right] \\ \textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{`senator'}\end{array}\right] \\ \textsc{obj}_\theta & \left[\begin{array}{ll}\textsc{pred} & \text{`gift'}\end{array}\right] \end{array}\right]$$

Model output

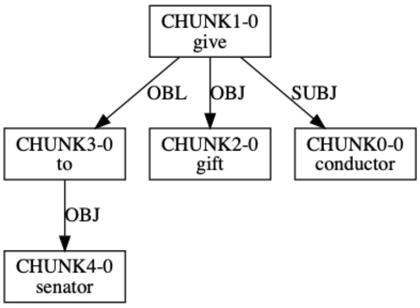


(192) The conductor gave a gift to the senator.

Prediction

$$\left[\begin{array}{ll} \textsc{pred} & \text{`give}\left\langle \textsc{subj},\ \textsc{obj},\ \left\{\textsc{obj}_\theta \mid \textsc{obl}\right\}\right\rangle\text{'} \\ \textsc{subj} & \left[\begin{array}{ll}\textsc{pred} & \text{`conductor'}\end{array}\right] \\ \textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{`gift'}\end{array}\right] \\ \textsc{obl} & \left[\begin{array}{ll}\textsc{pred} & \text{`to}\left\langle\textsc{obj}\right\rangle\text{'} \\ \textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{`senator'}\end{array}\right]\end{array}\right] \end{array}\right]$$

Model output



For all of the above sentences, the model output matches the prediction. The model uses partial feature specifications to link categories to potential grammatical functions, and to allow for grammatical function alternations, and this is able to replicate the theoretical analysis.

6.3.1.2 Korean

The model is intended to process input sentences from more than one language, using a single production set and lexically-specified combinatory constraints. The following sentences in Korean test that capability.

Sentences (193) and (194) show a transitive sentence with the two possible argument orders. In both cases, the processing outcome is predicted to be the same.

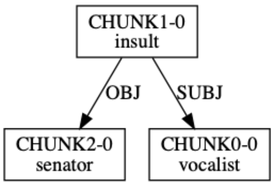
(193) *sengakkaka uywonul moyokhaysssta*
vocalist.SBJ senator.OBJ insulted

"The vocalist insulted the senator."

Prediction

$$\begin{bmatrix} \textsc{pred} & \text{`insult}\left\langle \textsc{subj},\ \textsc{obj} \right\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{`vocalist'} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`senator'} \end{bmatrix} \end{bmatrix}$$

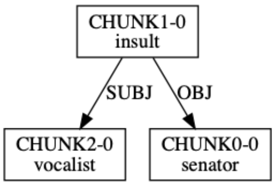
Model output



(194) *uywonul sengakkaka moyokhaysssta*
senator.OBJ vocalist.SBJ insulted

"The vocalist insulted the senator."

Model output



The model outputs have corresponding nodes and grammatical function arcs. However, the chunk names are different between the two examples, reflecting the order in which the nouns were processed.

Sentence (195) is a ditransitive sentence including an OBL argument. Here that argument is obligatorily marked with the particle *-ey* which assigns OBL grammatical function and indicates the thematic role of goal, or of location for stative verbs.

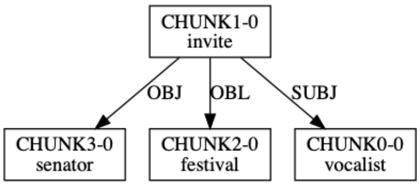
(195) *sengakkaka chwukceney uywonul chotayhaysssta*
vocalist.SBJ festival.LOC senator.OBJ invited

"The vocalist invited the senator to the festival."

Prediction

$$\begin{bmatrix} \textsc{pred} & \text{‘invite}\left\langle \textsc{subj},\ \textsc{obj},\ \textsc{obl} \right\rangle\text{’} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{‘vocalist’} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘senator’} \end{bmatrix} \\ \textsc{obl} & \begin{bmatrix} \textsc{pred} & \text{‘festival’} \end{bmatrix} \end{bmatrix}$$

Model output



Again, the model output matches the predicted analysis.

6.3.2 Complex sentences

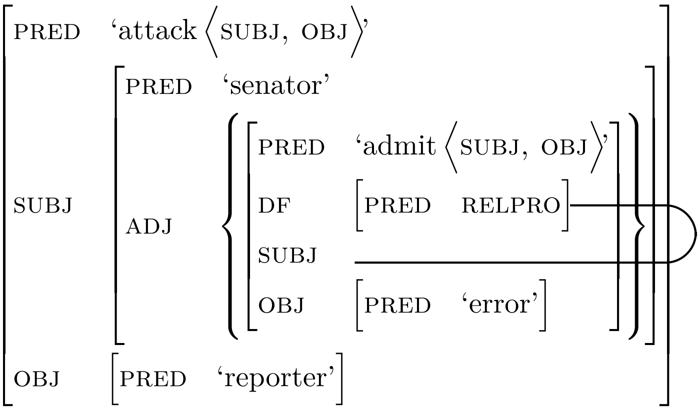
Complex sentences involve an embedded clause that provides a dependent grammatical function — argument or adjunct — to another PRED value.

6.3.2.1 English

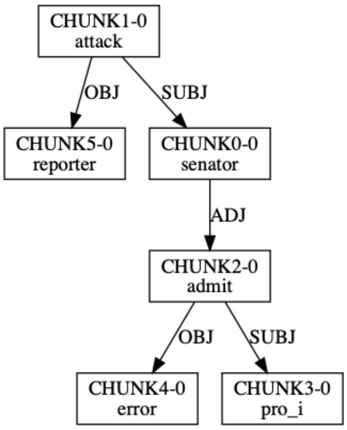
The first set of test items are subject- and object-relative clauses (SRC and ORC respectively) modifying the subject or object of a transitive verb.

(196) The senator who admitted the error attacked the reporter.

Prediction

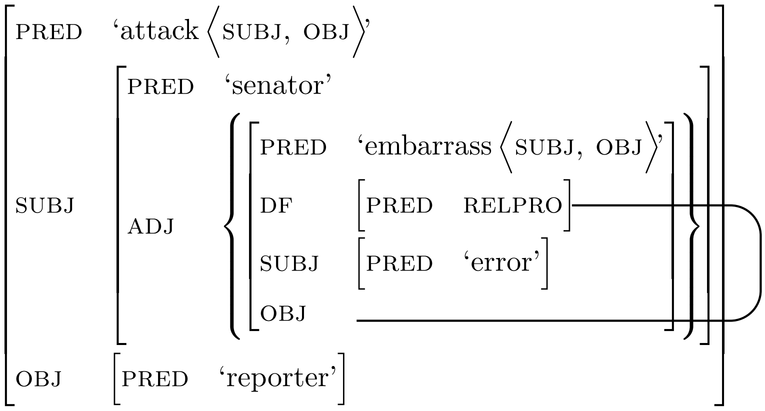


Model output

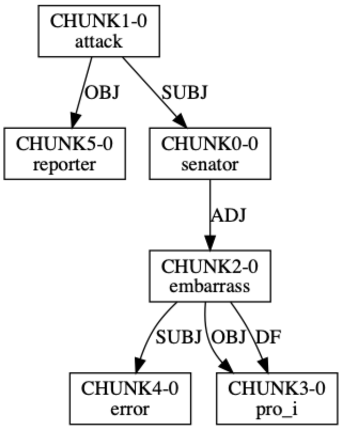


(197) The senator who the error embarrassed attacked the reporter.

Prediction

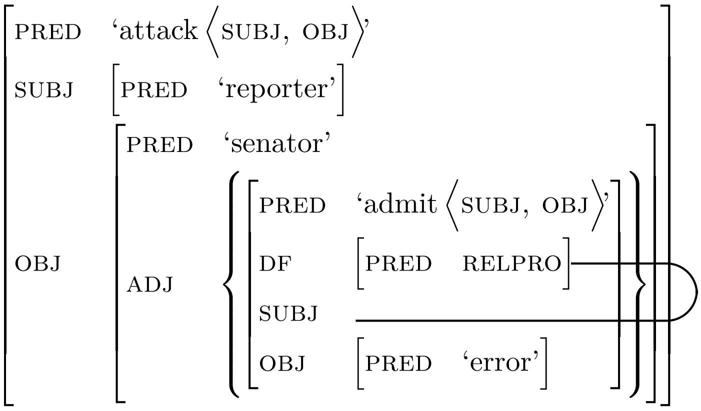


Model output

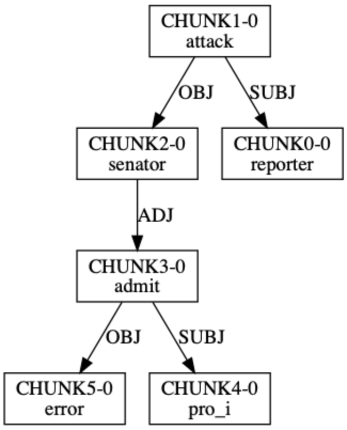


(198) The reporter attacked the senator who admitted the error.

Prediction

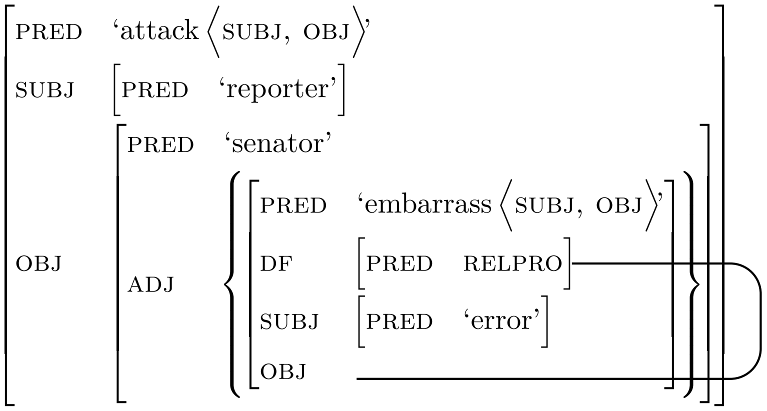


Model output

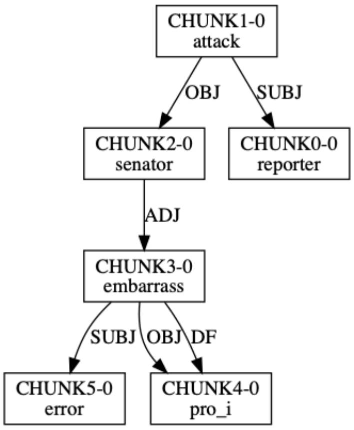


(199) The reporter attacked the senator who the error embarrassed.

Prediction



Model output



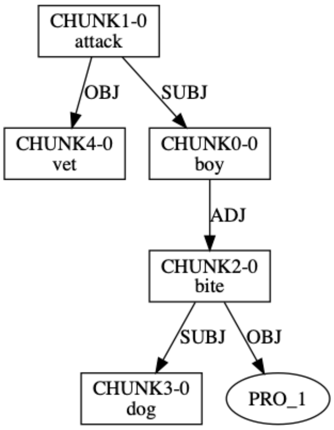
For all of the above sentences, where the left edge of the embedded clause is marked by a relative pronoun, the model output matches the prediction. The test items also include reduced relative clauses, where no complementiser or relative pronoun is present. In the next two examples, the reduced relative clause is the string *the boy the dog bit*.

(200) *The boy the dog bit* attacked the vet.

Prediction

$$\left[\begin{array}{ll}
\textsc{pred} & \text{`attack}\left\langle\textsc{subj},\ \textsc{obj}\right\rangle\text{'} \\
\textsc{subj} & \left[\begin{array}{ll} \textsc{pred} & \text{`boy'} \\ \textsc{adj} & \left\{\left[\begin{array}{ll} \textsc{pred} & \text{`bite}\left\langle\textsc{subj},\ \textsc{obj}\right\rangle\text{'} \\ \textsc{subj} & \left[\begin{array}{ll}\textsc{pred} & \text{`dog'}\end{array}\right] \\ \textsc{obj} & \end{array}\right]\right\}\end{array}\right] \\
\textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{`vet'}\end{array}\right]
\end{array}\right]$$

Model output

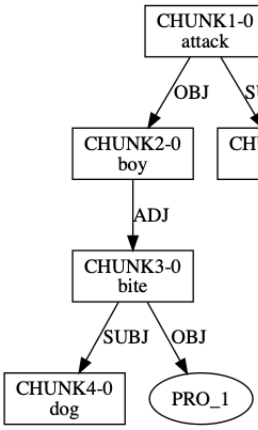


(201) The vet attacked *the boy the dog bit.*

Prediction

$$\begin{bmatrix} \textsc{pred} & \text{‘attack}\left\langle\textsc{subj},\ \textsc{obj}\right\rangle\text{’} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{‘vet’}\end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘boy’} \\ \textsc{adj} & \left\{\begin{bmatrix} \textsc{pred} & \text{‘bite}\left\langle\textsc{subj},\ \textsc{obj}\right\rangle\text{’} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{‘dog’}\end{bmatrix} \\ \textsc{obj} & \end{bmatrix}\right\} \end{bmatrix} \end{bmatrix}$$

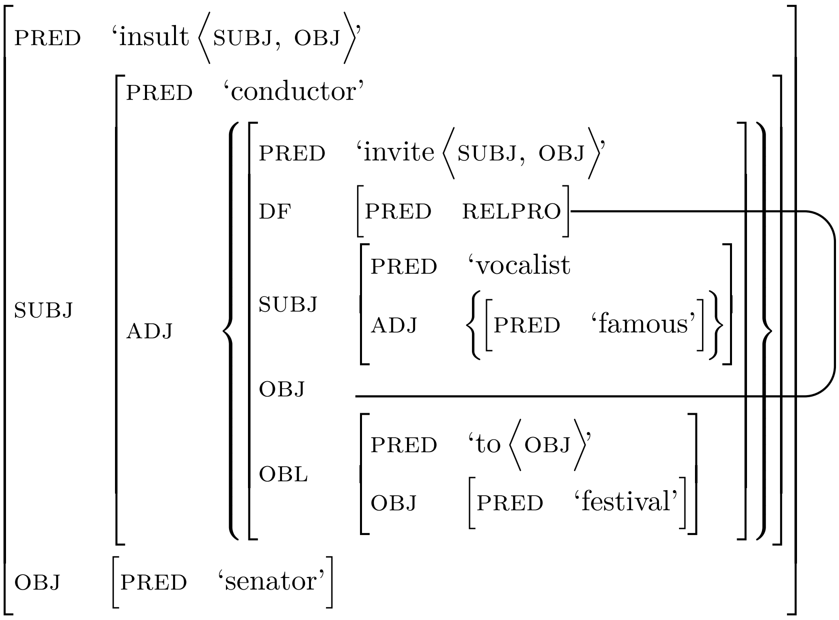
Model output



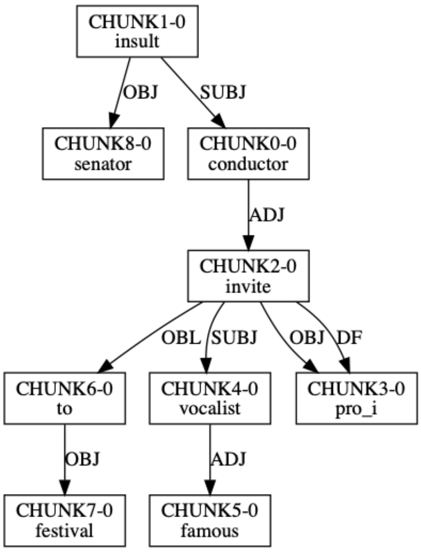
The test items also include an English version of the Korean complex sentences in Section 6.3.2.2.

(202) The conductor who the famous vocalist invited to the festival insulted the senator.

Prediction



Model output



6.3.2.2 Korean

Kwon et al. (2010) explored the role of semantic and syntactic factors in the processing of relative clauses. Six variants on a sentence were presented, taking three configurations of subject, object and modifier clause (203) and in each case having ORC and SRC variants of the modifier[6]. In examples (204)–(206), only one predicted f-structure is given for each ORC/SRC pair. Variables GF_1 and GF_2 in the f-structure each represent either SUBJ or OBJ depending on the clause type. A key is given for each example.

(203) a. [Mod S] O V

---

[6] The order of the matrix clause participants is *conductor – senator* in all six sentences, which means that in configurations (203a) and (203c) the conductor is insulting, whereas in configuration (203b) they are insulted.

b. [Mod O] S V

c. S [Mod O] V

(204) [Mod S] O V: Modifier is clause-initial, modifying the matrix subject.

a. ORC modifying matrix subject

yumyenghan sengakka-ka chwukceney chotayhan cihwuycaka
famous vocalist-SUBJ festival.LOC invited conductor.SUBJ
uywonul kongkongyenhi moyokhayssta
senator.OBJ publicly insulted

ORC: "The conductor who the famous vocalist invited to the festival publicly insulted the senator."

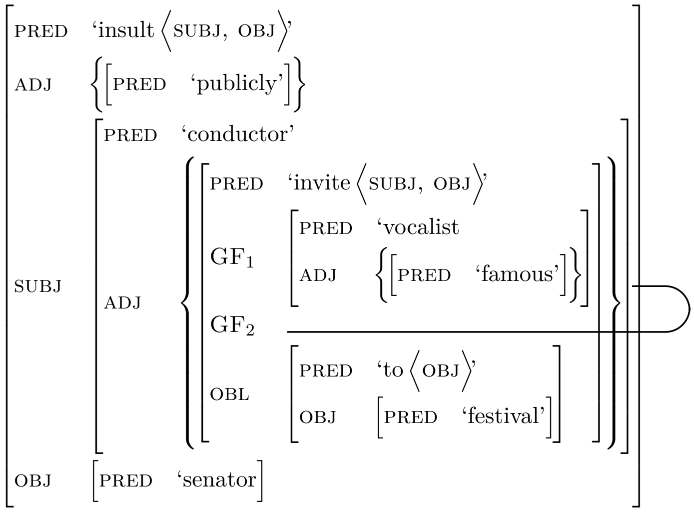
b. SRC modifying matrix subject

yumyenghan sengakka-lul chwukceney chotayhan cihwuycaka uywonul
famous vocalist-OBJ festival.LOC invited conductor.SUBJ senator.OBJ
kongkongyenhi moyokhayssta
publicly insulted

SRC: "The conductor who invited the famous vocalist to the festival publicly insulted the senator."

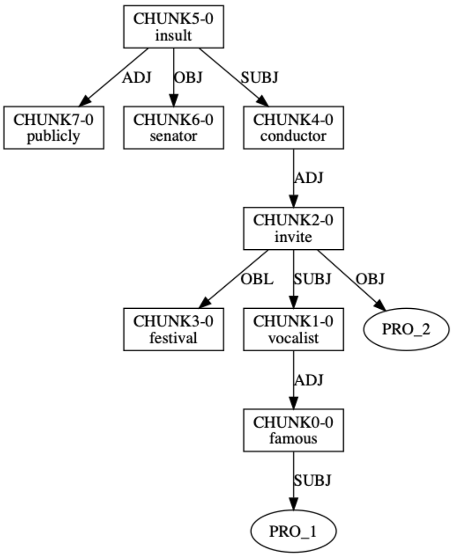
Prediction

ORC: $\mathrm{GF}_1 = \textsc{subj}$, $\mathrm{GF}_2 = \textsc{obj}$
SRC: $\mathrm{GF}_1 = \textsc{obj}$, $\mathrm{GF}_2 = \textsc{subj}$

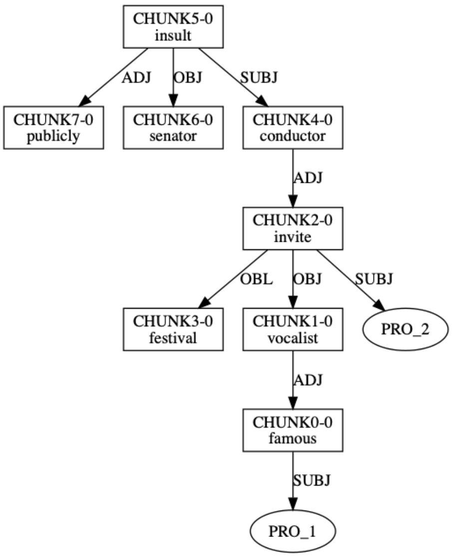


Model output

[ORC S] O V



[SRC S] O V



(205) [Mod O] S V: Clause-initial modifier of scrambled matrix object.

a. ORC modifying scrambled matrix object

yumyenghan sengakka-ka chwukceney chotayhan cihwuycalul
famous vocalist-SUBJ festival.LOC invited conductor.OBJ
uywoni kongkongyenhi moyokhayssta
senator.SUBJ publicly insulted

ORC: “The senator publicly insulted the conductor who the famous vocalist invited to the festival.”

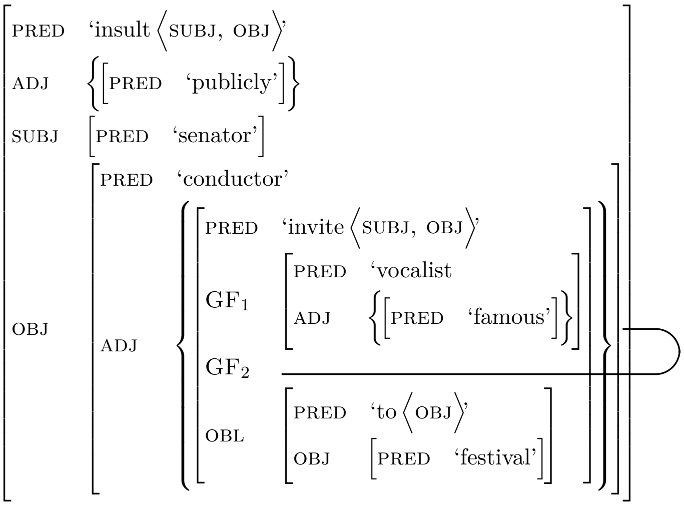
b. SRC modifying scrambled matrix object

yumyenghan sengakka-lul chwukceney chotayhan cihwuycalul uywoni
famous vocalist-OBJ festival.LOC invited conductor.OBJ senator.SUBJ
kongkongyenhi moyokhayssta
publicly insulted

SRC: “The senator publicly insulted the conductor who invited the famous vocalist to the festival.”

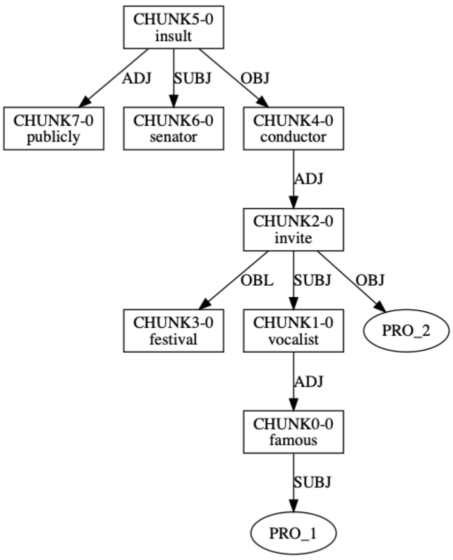
Prediction

ORC: $\mathrm{GF}_1 = \textsc{subj}$, $\mathrm{GF}_2 = \textsc{obj}$
SRC: $\mathrm{GF}_1 = \textsc{obj}$, $\mathrm{GF}_2 = \textsc{subj}$

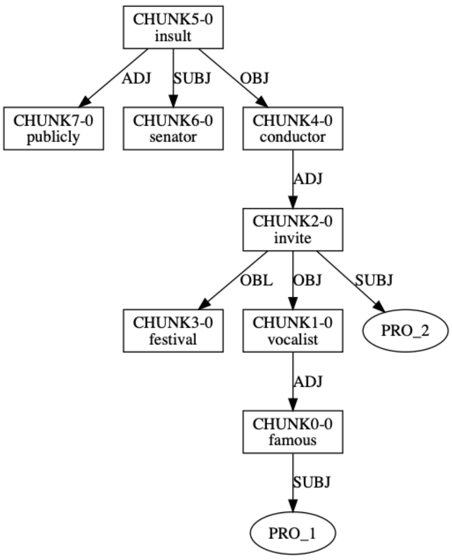


Model output

[ORC O] S V



[SRC O] S V



(206) S [Mod O] V: Modifier is the second constituent in the sentence, after the matrix subject and modifying the matrix object.

a. ORC modifying matrix object in canonical position

yumyenghan	*cihwuycaka*	*sengakka-ka*	*chwukceney*	*chotayhan*	*uywonul*	*kongkongyenhi*	*moyokhayssta*
famous	conductor.SUBJ	vocalist-SUBJ	festival.LOC	invited	senator.OBJ	publicly	insulted

ORC: "The famous conductor publicly insulted the senator who the vocalist invited to the festival."

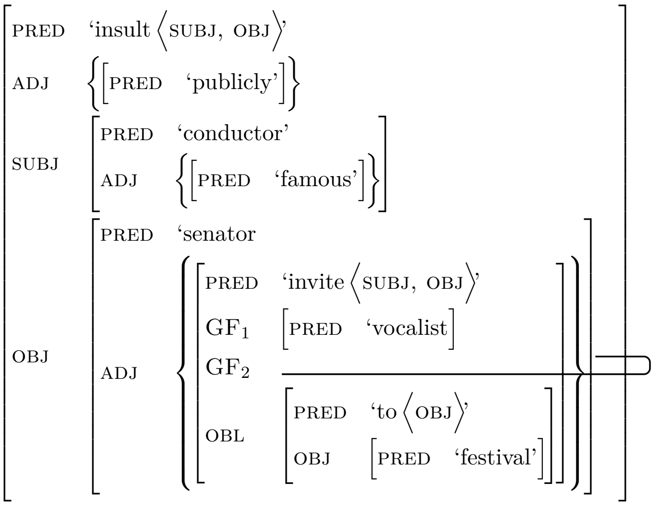
b. SRC modifying matrix object in canonical position

yumyenghan	*cihwuycaka*	*sengakka-lul*	*chwukceney*	*chotayhan*	*uywonul*	*kongkongyenhi*	*moyokhayssta*
famous	conductor.SUBJ	vocalist-OBJ	festival.LOC	invited	senator.OBJ	publicly	insulted

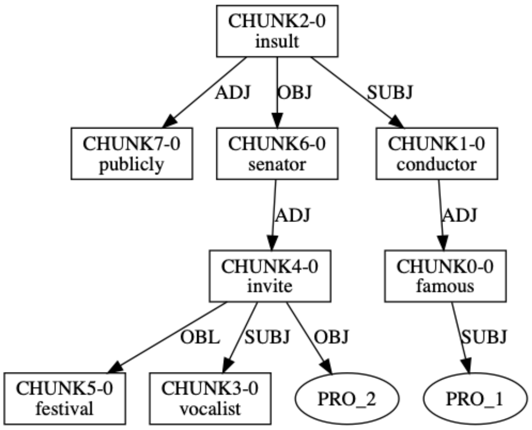
SRC: "The famous conductor publicly insulted the senator who invited the vocalist to the festival."

Prediction

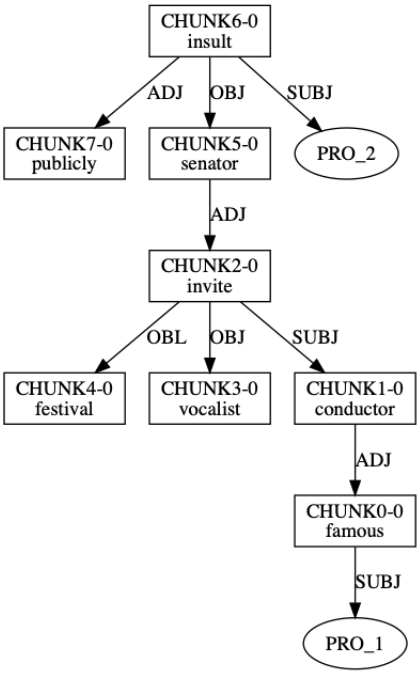
Matrix object is modified
ORC: GF_1 = SUBJ, GF_2 = OBJ
SRC: GF_1 = OBJ, GF_2 = SUBJ



Model output: S [ORC O] V



Model output: S [SRC O] V



For the sentence pairs [Mod S] O V (204), [Mod O] V S (205), and the ORC variant of S [Mod O] V (206a), the model output matches the prediction. However, the SRC variant of S [Mod O] V (206b) does not generate the intended reading (207a), but instead generates a degraded reading, an approximate rendering of which is given in (207b).

(207) a. The famous conductor publicly insulted the senator who invited the vocalist to the festival.

b. * “Publicly insulted the senator who the famous conductor invited the vocalist to the festival.”

This is due to a garden path effect: the first three NPs in the sentence are marked for different grammatical functions, and so there is no reason to expect that they should not all be in the same clause.

(208) *yumyenghan cihwuycaka sengakka-lul chwukceney …*
famous conductor.SUBJ vocalist-OBJ festival.LOC …
“The famous conductor … the vocalist to the festival…”

$$\begin{bmatrix} \textsc{pred} & \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{`conductor'} \\ \textsc{adj} & \left\{ \begin{bmatrix} \textsc{pred} & \text{`famous'} \end{bmatrix} \right\} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`vocalist'} \end{bmatrix} \\ \textsc{obl} & \begin{bmatrix} \textsc{pred} & \text{`festival'} \end{bmatrix} \end{bmatrix}$$

When *chotayhan* ‘invited.ADN ⟨SUBJ, OBJ, OBL⟩’ is introduced, there is no reason for it not to fill the empty PRED position and because the clause could be functioning to modify a noun that would serve as a temporal adjunct (e.g. *the day the famous conductor invited* *the vocalist to the festival*), there is no inherent structural difficulty with the sentence. Once *uywonul* 'senator.OBJ' is introduced as the noun modified by the clause, structural processing can continue but there is no explicit subject for the matrix clause, as the SUBJ-marked noun has been processed within the modifier clause.

This model is building initial structure and does not address the issue of repair. However, modelling prosodic support for processing complex structure is discussed in Section 6.4.2.

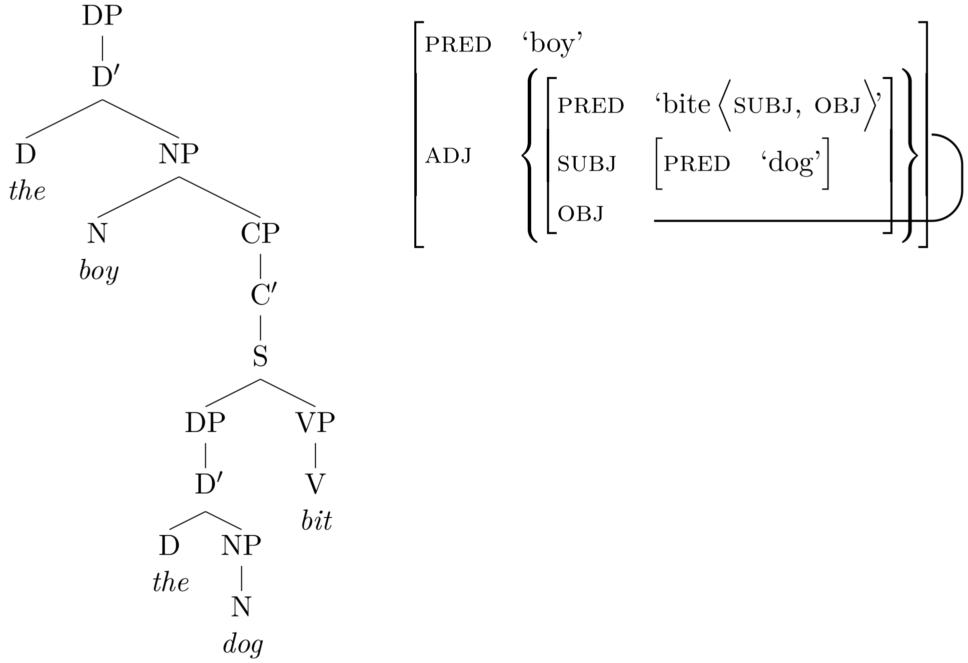
6.4 Testing the model: specific predictions

Having demonstrated that the model can successfully generate syntactic structure for simple sentences and single-embedded complex sentences, I move on to test some specific predictions of the incremental theory.

6.4.1 Context variations for identical input strings

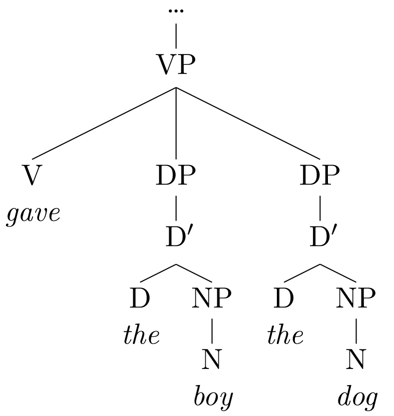
In Section 6.3.2 I introduced sentences containing the string *the boy the dog bit.* In transitive sentences and without a prosodic break, the string is interpreted as a noun phrase modified by a reduced relative clause (209), where the clause *the dog bit* is interpreted as a relative clause adjunct.

(209)



However, following prediction 3 from Chapter 3, structure will be preferentially built at an available governable grammatical function, rather than at an adjunct site. Thus following the ditransitive verb *gave*, it is predicted that *the dog* will be preferentially interpreted as the OBJ_θ grammatical function of *gave* rather than as an adjunct to *the boy*, causing a garden path effect when *bit* is subsequently encountered (210).

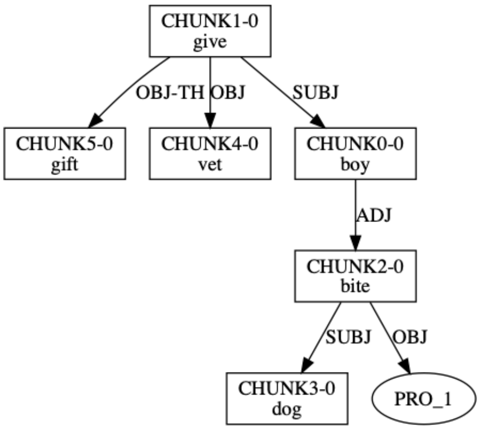
(210) The vet gave *the boy the dog*...



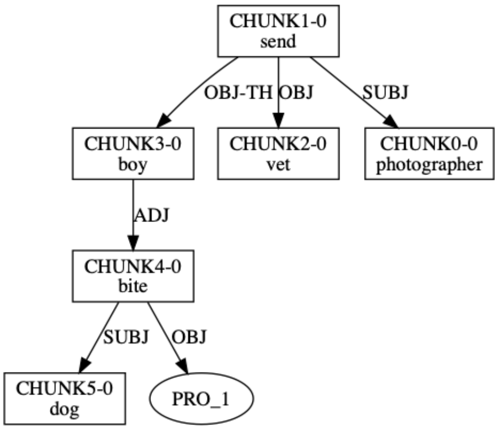
$$\left[\begin{array}{ll} \vdots & \vdots \\ \textsc{pred} & \text{`give}\left\langle \textsc{subj},\ \textsc{obj},\ \left\{\textsc{obj}_\theta \mid \textsc{obl}\right\}\right\rangle\text{'} \\ \textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{`boy'}\end{array}\right] \\ \textsc{obj}_\theta & \left[\begin{array}{ll}\textsc{pred} & \text{`dog'}\end{array}\right] \end{array}\right]$$

These predictions are successfully replicated by the model. When *the boy* is the SUBJ (211) or $\textsc{obj}_\theta$ (212) argument of *gave*, the string *the boy the dog bit* is processed as a reduced relative clause. However, when *the boy* is the OBJ argument of GAVE, *the dog* is processed as the $\textsc{obj}_\theta$ argument, leading to an incoherent structure (213).

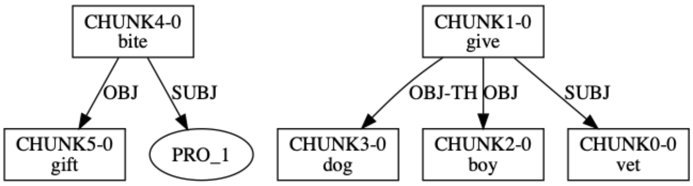
(211) *The boy the dog bit* gave the vet a gift



(212) The photographer sent the vet *the boy the dog bit.*

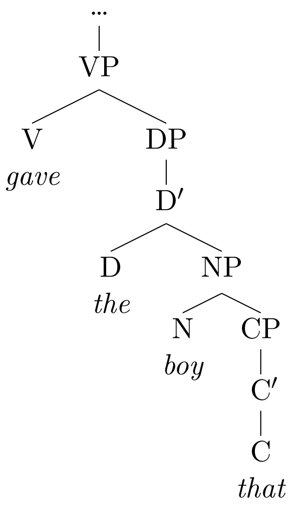


(213) The vet gave *the boy the dog bit* a gift.



This garden path effect is cancellable if a complementiser is present, as shown in example (214) repeated from example (138) in Section 3.3.3.1, p. 91. The complementiser *that* introduces an underspecified f-structure whose PRED value is constrained to be [CAT V +] within the f-structure headed by *boy*. Prediction 1 of the theory is that attachment may not take place outside the newly-introduced f-structure until its PRED value is present. Because of this, the OBJ$_\theta$ argument of *give* is not accessible and so even though *the boy* is not at this stage providing a governed grammatical function within the relative clause, the theory predicts it must be attached here rather than in the matrix clause.

(214) The vet gave *the boy* that...



$$\left[\begin{array}{ll} \vdots & \vdots \\ \mathrm{PRED} & \text{‘give}\left\langle \mathrm{SUBJ},\ \mathrm{OBJ},\ \left\{\mathrm{OBJ}_\theta \mid \mathrm{OBL}\right\}\right\rangle\text{’} \\ \mathrm{OBJ} & \left[\begin{array}{ll} \mathrm{PRED} & \text{‘boy’} \\ \mathrm{ADJ} & \left\{\left[\begin{array}{ll} \mathrm{PRED} & \\ \mathrm{COMPFORM} & \text{‘that’}\end{array}\right]\right\} \end{array}\right] \\ \mathrm{OBJ}_\theta/\mathrm{OBL} & \end{array}\right]$$

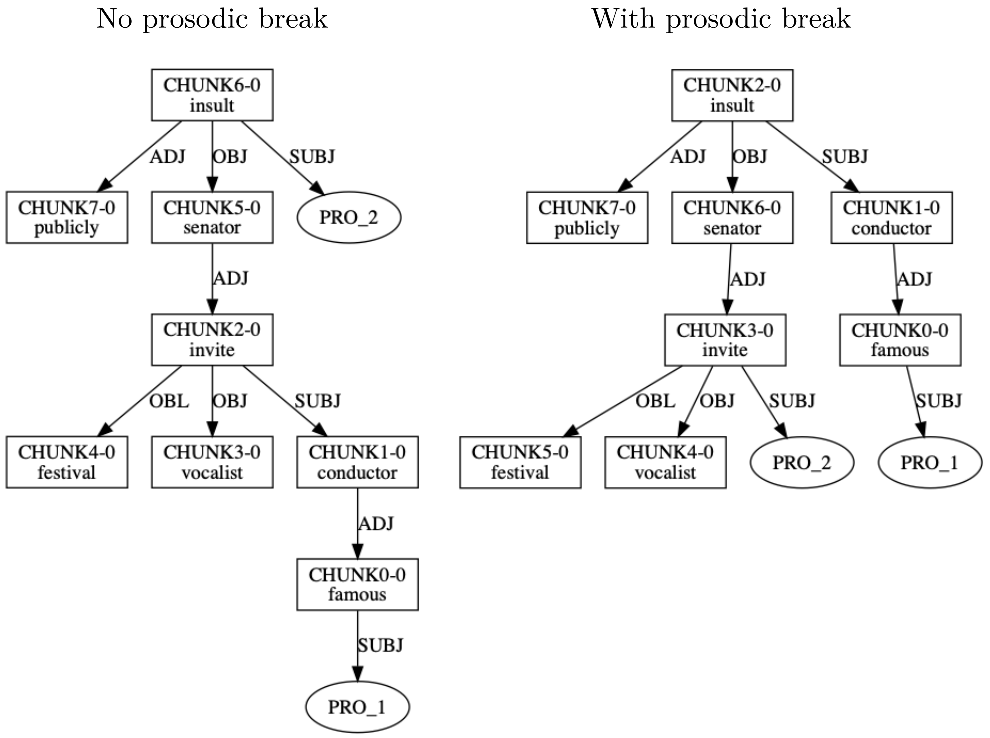
6.4.2 The impact of prosody

In Section 6.3.2.2 test sentences from Kwon et al. (2010) were modelled. The sentence structures are complicated and difficult to process: the authors themselves note that the total reading times for the sentences are greatly in excess of what would be expected. The model was able to generate the intended meaning for five of the six sentences, but failed for the variant where the sentence begins with the matrix subject, which is followed by an SRC modifying the matrix object (206b). The hypothesised reason for this is a garden path.

If a prosodic break signals the right edge of a constituent and, by inference, the left edge of the next one, the incremental theory predicts that the processing of sentence (206b) can be facilitated by introducing a prosodic break after the matrix subject. The impact of this is shown in (215) where, with the prosodic break, the model successfully generates the intended reading.

(215) [*yumyenghan cihwuycaka*] || [*sengakka-lul chwukceney chotayhan*]
[famous conductor.SUBJ] || [vocalist-OBJ festival.LOC invited]
uywonul kongkongyenhi moyokhaysta
senator.OBJ publicly insulted

SRC: "[The famous conductor] publicly insulted the senator [who invited the vocalist to the festival]."


No prosodic break

With prosodic break

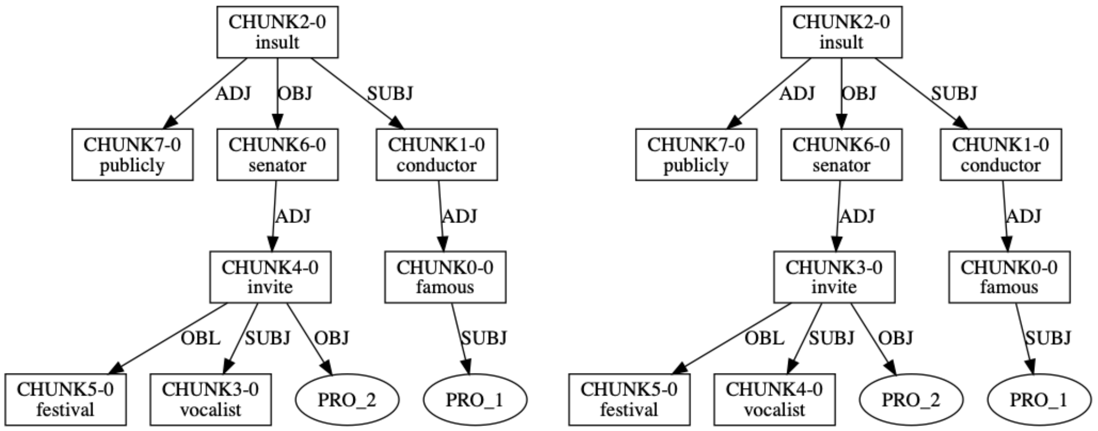
The incremental theory predicts that a prosodic break after the matrix subject in the corresponding ORC (216) will have a neutral impact on the processing outcome. In this sentence the second NP introduces a second SUBJ-marked constituent and so a subordinate clause is introduced at that point in preference to an incoherent clause with two SUBJ grammatical functions. The model output replicates this prediction: introducing a prosodic break does not change the structure that is generated.

(216) ORC modifying matrix object in canonical position

[*yumyenghan cihwuycaka*] || [*sengakka-ka chwukceney chotayhan*]
[famous conductor.SUBJ] || [vocalist-SUBJ festival.LOC invited]
uywonul kongkongyenhi moyokhaysta
senator.OBJ publicly insulted

ORC: "The famous conductor publicly insulted the senator who the vocalist invited to the festival."

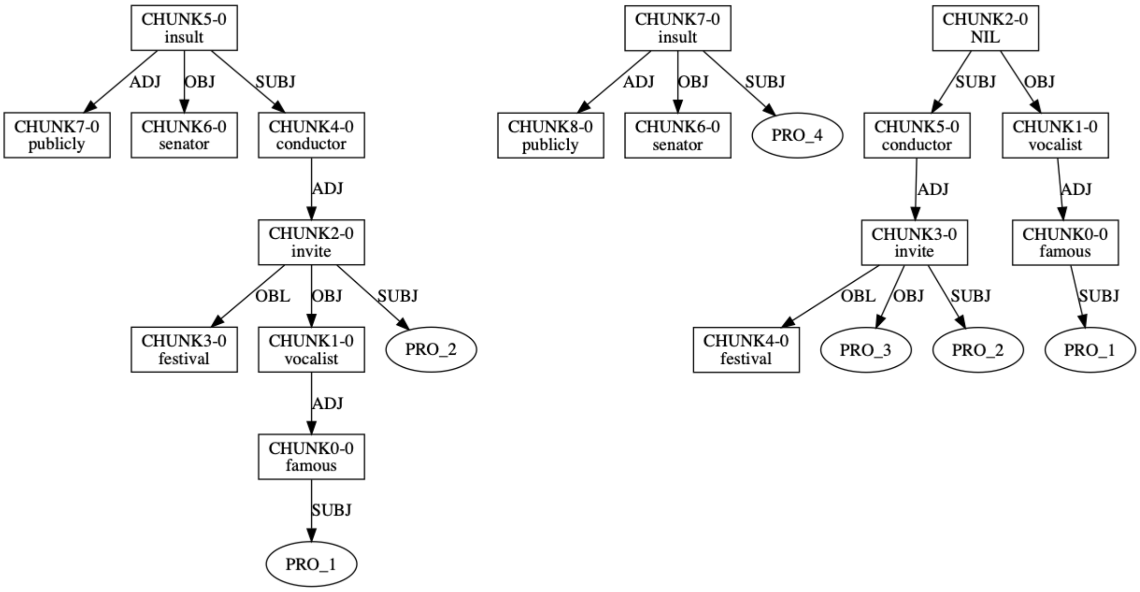
No prosodic break — With prosodic break



Introducing a prosodic break after the first NP constitutent in the other sentence variants is predicted to degrade the sentences: this prediction has been confirmed informally by Korean native-speaker informants but has not been formally tested. An example of the impact on the SRC variant of [Mod S] O V sentence (204b) is shown at (217).

(217) [*yumyenghan sengakka-lul*] || [*chwukceney chotayhan*] *cihwuycaka*
[famous vocalist-OBJ] || [festival.LOC invited] conductor.SUBJ
uywonul kongkongyenhi moyokhayssta
senator.OBJ publicly insulted
SRC: "The conductor who invited to the festival…… publicly insulted…the famous vocalist…the senator."

No prosodic break — With prosodic break



Introducing a prosodic break in the middle of the modifying clause results in an incoherent analysis with two unconnected clauses.

6.5 Discussion of the findings

In this section I review the operation of the model, summarising its successes in replicating human performance, and setting out its limitations. I then discuss issues related to modelling the incremental theory within the design constraints of ACT-R, and issues related to modelling the incremental theory more generally.

6.5.1 What worked?

In Section 6.3, I demonstrated how the model builds sentence structure using lexically-specified constraints. The aim was to do this for two typologically different languages, English and Korean, using a single set of productions. The single production set in the model is able to build monoclausal sentences with verbal heads of valencies that use various combinations of OBJ, OBJ_θ, and OBL arguments.

Feature-based representations of syntactic category and grammatical function were used successfully to model the dative alternation of *give* in English. The model replicates pro-drop where licensed by the grammar, resulting in different treatments of missing arguments in English and Korean. Pro-drop is used in the treatment of reduced relative clauses in English.

The model can also process complex syntax, handling embedded clauses with (English) or without (English, Korean) an explicit complementiser. Where embedding is not triggered by an explicit complementiser, the left edge of an embedded clause can be signalled by a constituent that otherwise cannot attach, or which would produce an incoherent structure. However, where there was no clear signal that embedding should begin, the model produced a degraded reading of the sentence, different to the reading intended by the producer of the sentence.

Embedding can also be signalled prosodically: the model includes a treatment of a “prosodic break” that works identically in both English and Korean, signalling the right edge of a constituent and causing a processing action to be taken that otherwise would not have been the preferred decision.

Section 6.4 then moved on to the modelling of specific phenomena. In Section 6.4.1, I presented data on the context-dependent processing of reduced relative clauses (RRCs) in English. Where the head of the RRC is the object of a ditransitive verb, a garden path effect is reported by human participants, with the RRC subject being treated as the OBJ_θ argument of the matrix clause. However, this garden path is cancellable if the relative clause is introduced by an overt complementiser. The incremental theory elaborated in Chapter 3 accounts for this by the different operation of universal constraints on structure-building depending on the lexical information which is available, and and the model successfully replicates this account. Structures containing RRCs are generated where no garden path is seen, and a degraded structure is produced for the sentence that results in a garden path for human subjects.

In Section 6.4.2, data were presented from a model that introduced prosodic support alongside complex syntax in Korean. This model was able to reproduce human performance for the complex sentence that had not been successfully processed in Section 6.3.2.2. The incremental theory predicted that introducing a prosodic break at the same ordered (not hierarchical) position with the sentence would have either no impact on processing, or would degrade the processing of the sentence dependent on the hierarchical structure of the sentence, and the model replicated this effect.

6.5.2 Limitations of the model

The model performed well for the tasks that it was set, but its operation is based on restricted, core grammars of both English and Korean, and the test items considered a restricted set of phenomena within both languages. The single production set was able to deliver results, and for many of the goal states, the processing is identical regardless of word-order constraints. However, one result of the significant typological differences between English and Korean is that within the `create` goal state, the productions dealing with attachment once a verbal predicate has been processed are only used for English sentences. Thus it could be argued that there are still distinct subsets of language-specific knowledge within ACT-R's procedural memory, rather than language-specific knowledge being contained in lexical specifications within the declarative memory. One way to test this would be to test the model with languages such as German or Czech, where there is still a structural element to allocating grammatical function — e.g. the start and end of the verbal bracket in German (Durrell, 2002), the fixed position of clitics after the first sentence constituent in Czech (Janda and Townsend, 2000) — but also more freedom in word order without the rigorous head-finality of Korean.

The model's focus on initial structure-building has excluded repair processes. The worked example of building a Korean sentence using the incremental theory, presented in Section 3.4, ended with three possible choices of a final structure that had subtle differences in meaning. Situations such as this are likely to be the norm, rather than the exception. As discussed in Chapter 2, "repair" may take place at any level of mental representation, including at a more abstract level that than the initial structural representation, and these repaired abstract representations may coexist in memory with unrepaired, lower-level representations. However, this does not preclude rapid repair taking place during early structure building, but the model is not yet capable of this.

The assumed position of language processing in wider cognition, discussed in Chapter 5 includes the early addition of new participants to the discourse representation, before their semantic role is computed. This assumption in the theory allows memories associated with new discourse participants to interface with inferential processing, and to influence structural choices during compositional processing, but this is not yet in place in the model.

The focus of the model was on structural outputs rather than timings. A multibuffer was used to manage the consequences of embedding, rather than repeated retrievals of structural chunks moderated by either a processing stack or or by specific structural types that allow for search cues to differentiate between matrix, embedded, and gapped clauses. The overall modelled time course of a processing cycle results in processing at a slower rate than human subjects process written or aural input. Because the only memory retrievals in the model are for lexical entries, the time-course depends principally on selection of productions, resulting in a 50ms granularity of variation. This also does not reflect experimental data. Thus to be of use for modelling time-course data as well as testing a structure-building theory, considerable work is needed on the model architecture.

The model's use of a multibuffer for embedding was able to match human performance for singly-embedded structures, but does not match human performance for multiply-embedded structures. It offers neither the over-accurate processing of multiple embedding that could be seen by using a stack, nor the sensitivity to structure and variations in processing ease of the retrieval-based model in Lewis and Vasishth (2005). However, the reliance of the theory on structural chunks that are not restricted to binary branching conflicts with the ACT-R treatment of chunks that are modified after they are retrieved.

6.5.3 ACT-R intrinsic challenges

Perhaps the most significant challenge for language modelling in ACT-R is the architecture's treatment of modified chunks. Once a chunk has been retrieved and its slot values altered, the new version of the chunk coexists in declarative memory with the previous, unaltered version. This results in increased competition between chunks when subsequent retrievals are made. If I have processed chunk B a number of times such that versions B, B' and B'' are present in memory, and am seeking to attach to chunk A', which was processed before B, but has similar empty slots to chunk B', all other things being equal, chunk B' will be retrieved because it was more recently activated: this challenge was encountered in the early stages of this project, and was also reported on by Ball (2012a). Allowing a newly-released copy chunk to merge with its parent is one possible solution, but this requires changing the central operation of ACT-R declarative memory and may make the model too powerful. Restricting merger to chunks where the content of slots has increased, but no individual slot has lost content or changed its value, could be a way to enable the operations needed for language processing without increasing the power of the architecture unnecessarily. Exploring and resolving this issue is left for future work.

A broader question for the modelling of the LFG incremental theory is the practical challenge of modelling multiple simultaneous processes: initial structure building, complex calculations, argument assignment, proposition calculation, two-way interaction with inferential processing. Based on empirical data, it is difficult to reach a conclusion other than that the processing of one word begins before the preceding word is complete. Modellers must find a way to specify the boundaries between different tasks and describe how information passes between the tasks in both directions. In the LFG theory, projection of levels is a pipeline, with one level being derived from the previous one, and this brings clarity to the modelling task. However, that does not guarantee that any feedback loops (e.g. constraints or preferences arising from the discourse structure or s-structure influencing a decision on which lexical entry to choose for a homonym) will follow the same pipeline in the reverse direction. A first step might be to isolate a specific task, such as the modelling of incremental s-structure growth from available f-structural information, and explore the issues that arise in practice. Again, this is left for future work.

6.6 Summary

This chapter presented the details of an ACT-R model that has been developed from an incremental theory of LFG, using a single production set to process English and Korean text, and has compared its performance to reported judgements from native speakers. The model is able to build monoclausal and biclausal sentence structures, and has matched theoretical predictions and human performance data in the processing of strings that can trigger a context-dependent garden path, and in modelling the impact of prosodic support on the processing of complex syntax.

The model has been tested on a core grammar of the two languages using a restricted set of test sentences, and the limitations and implications of this have been explored, including the impact of ACT-R design choices. A number of potential avenues for future work have been identified, including the modelling of time-course as well as structural data, and the expansion of the model into other steps of the overall language processing pathway.

Chapter 7

Conclusions

This thesis responds to two of the challenges posed to formal linguistics by Ferreira (2005): to engage with developments in cognitive science, and to integrate semantics and phonology with formal treatments of syntax. Here I set out the findings of my research in relation to those challenges, and discuss potential avenues for future research.

7.1 Summary of findings

In Chapter 2, I motivated the development of an incremental approach to LFG by considering the representations of language structure in theories of grammar that specifically address incremental sentence growth, and noting that they may not be adequate to develop accounts of grammar in processing. I reviewed evidence on the role of grammar in processing and concluded that, while syntax has a pervasive impact on the processing pathway, a full account of processing phenomena must be able to include the impact of phonological, semantic and discourse factors as well as syntactic structure. However, the interactions between these factors add significant complexity to an account, which in turn makes it difficult to test the specific impact of syntactic constraints. The role of syntax is clearest at the earliest stages of structure-building, and from this I concluded that a first attempt to develop a model of structure-building should limit itself strictly to those early stages. I also reviewed evidence about the interaction between prosody and structure-building, noting that prosodic boundaries can have a very early impact on attachment choices, and concluded that a model of the early stages of parsing should attempt to include a representation of prosody.

In Chapter 3, I introduced LFG which, as a modular theory of the relationship between language form and meaning, is well-placed for the construction of integrated accounts of the syntactic, semantic, discourse and phonological aspects of language. The LFG architecture is directional, with an ordered projection of the levels of representation, but an incremental theory allows the precise specification and testing of hypotheses about immediate relationships between non-adjacent levels of representation, provided that those relationships consider the information available at different points in the sentence. Having introduced canonical, declarative LFG, I extended the theory to address incremental sentence growth, looking in detail at the role of category and functional constraints on structure building. When applied incrementally, the universal constraints of LFG make specific predictions about constraints and preferences in structure-building, which can then be tested against human data.

An incremental approach also allows the language-specific categorial and functional constraints that are usually described in phrase-structure rules to be included in lexical specifications. This in turn allows structure to be built bottom-up based on lexical constraints, without requiring a separate parsing algorithm or discrete knowledge of phrase structure as is the case with Categorial Grammar, or with models representing surprisal-based theories. A consequence of this is that any model of structure-building should be able to work cross-linguistically using only language-specific lexicons. I tested the incremental theory on paper, using language input from Korean, and identifed specific issues relating to choice points during parsing.

In Chapter 4, I introduced the ACT-R cognitive architecture and briefly described its structure and function. I reviewed published models of language processing in cognitive architectures: all are seeking to replicate human performance in language processing, but some aim to fit detailed time-course variations linked to sentence structure, whereas others seek to deliver an accurate representation of sentential content at a speed that matches the rate at which language input is perceived. I noted the common assumptions made about the level and nature of additional working memory capacity within ACT-R, and the consequences of this for cross-linguistic models. I also noted the practical impact and risks that parallel representations of syntactic structure would introduce to the model's operation.

In Chapter 5, I set out the basis for a cognitive model of language structure-building constrained by the incremental theory of Chapter 3. Taking into account evidence on the processing pathway and the timings at which the impacts of factors other than syntactic constraints are seen, I proposed that the model should cover the earliest stages of structure-building, thus removing the need to make assumptions about the non-syntactic factors. I introduced a single representation of syntactic structure, the F-representation, which combines elements of LFG c-structure and f-structure, and which allows an LFG c- and f-structure analysis to be retrieved, but which enables the grammatical constraints to be applied without the need to synchronously update two parallel representations.

I set out how the model approaches structural ambiguities and argument alternations, including the use of underspecified feature representations for category and grammatical function constraints. I also motivated the assumption of additional working memory capacity in the model, whilst keeping this capacity as restricted as possible, and described how the model should govern a processing cycle and handle issues related to the proliferation of copies of memory chunks.

Then in Chapter 6, I introduced the model that I have developed, describing the structure of declarative memory, the production set used in procedural memory, and the processing pathway used to integrate successive words into the language representation. I then presented the outputs of the model. I demonstrated that the model can build structure from monoclausal and complex sentences in English and Korean, and that equivalent sentences in the two languages result in analogous structural representations despite very different constituent structures. I also tested the predictions of the theory regarding the context-dependent processing of a particular string, and the impact of prosodic breaks on the processing of complex syntax. I showed that the model replicates theoretical predictions and human performance when considering the impact of sentential context on processing. I also showed that the model's treatment of prosodic breaks results in facilitation or degradation of processing as predicted, depending on the relative alignment of prosodic breaks and the edges of embedded clauses.

7.2 Questions for future research

The incremental theory and the model presented here are a first step in using LFG incrementally. They demonstrate that an incremental theory can represent the constraints from declarative LFG, that these can be modelled using a cognitive architecture, and that language-specific constraints on syntactic structure building can be treated lexically, rather than requiring distinct knowledge of phrase structure. The model's fit to human performance considered structural representations rather than time-course variations, but the ability to deliver an accurate representation at least some of the time is a prerequisite of any model of language processing.

Below I identify three areas where the assumptions in my model removed the need to consider theoretical difficulties, or where it seems likely that the limitations of the model will need to be addressed explicitly.

The first of these is a more detailed elaboration of the incremental theory of LFG. Logical next steps would be to model the incremental growth of s-structure with the assignment of event participants to the argument positions of a predicate, considering the role of alternations such as passivisation and applicative constructions, and their relation to the alternate sets of grammatical functions generated by the model. Another question to explore would be the relationship between initial structure-building and the discourse. The processing pathway in Chapter 5 and the Partial Dynamic Semantics theory of Haug (2014) assume that new participants are introduced rapidly to the discourse representation once a lexical entry has been retrieved. Exploring the implications of this for the incremental theory, and the impact of maintaining a discourse representation on the activation level of individual ACT-R memory chunks, is potentially a productive research topic.

A second future research area would be to extend the scope of the cognitive model, either in terms of its grammatical coverage, or to test its structure-building capabilities with other languages. Although the constraints in English and Korean are very different, word order plays a significant role in both, steering the assignment of grammatical functions in English, and strictly determining the right edge of constituents in Korean. It would be interesting to explore how languages where word order is less prominent might be modelled, and what the implications are for the current model's assumptions on how incomplete constituents are maintained in working memory.

Constructing semantic representations from syntactic structural representations is another interesting modelling challenge: a first attempt at this might use F-representations as the input, rather than build a multilevel model that processes the input string. The existence of more than one type of representation enables hypotheses to be tested about repair processes, e.g. following garden path sentences. A common assumption is that repairs take place at the level of syntactic structure, but a model that maintains distinct syntactic and semantic representations might choose to repair only the semantic representation, leaving the syntactic representation unchanged. Empirical evidence on the nature of search cues for different purposes would be important in informing this work.

This brings us to the third question, which relates to ACT-R's treatment of declarative memory chunks. The model as it stands is not capable of replicating fine-grained time-course variations, because of its division of labour between working memory (represented by buffers) and declarative memory. Modelling LFG requires a single head to have multiple dependents. Because all of these are not attached simultaneously, the representation of a head in memory is dynamic. ACT-R declarative memory holds snapshot copies of a dynamically changing memory chunk at each point that the chunk was released back into declarative memory. The only way to avoid this is to keep a chunk active in working memory, but this has consequences for working memory capacity and prevents the modelling of activation decay processes. However, the proliferation of copy chunks inhibits the construction of accurate structural representations.

Changing a fundamental property of ACT-R could have significant implications for modelling of other psychological tasks. However, making a change only for language processing carries an implicit claim about language-specific cognitive processes, whereas much work in ACT-R to date has shown that no cognitive assumptions outside general cognitive processes are necessary in order to replicate human performance. Ultimately it may be necessary to consider a different cognitive architecture instead of, or alongside, ACT-R to find a satisfactory solution.

Appendix A

LFG grammar fragments

A.1 English

The phrase-structure rules below, used in the thesis, are a subset of those proposed by Falk (2001). Categories in the grammar are ADJ, ADV, C, D, I, N, P, S, V.

The rules make use of variables, instantiated as shown in Table A.1.

Table A.1: Variables used in phrase structure rules

Variable	Permitted instantiations
A	ADJ, ADV
X	ADJ, ADV, N, P, V (lexical categories)
Z	IP, S

A.1.1 Lexical precedence rules

(218)
a. X^0 initial
b. $\text{DP} \prec \text{PP}$
c. $\text{PP final} \vee \Big(\text{PP} \prec \{\text{CP}|\text{IP}\} \wedge \{\text{CP}|\text{IP}\} \text{ final}\Big)$
d. SPEC initial
e. $\text{AP} \prec \Big\{\text{NP}|\text{AP}|\text{PP}\Big\}$
f. $\text{OBJ} \prec_f \text{OBJ}_\theta$

A.1.2 Immediate dominance rules

Root categories are CP, IP, S.

A.1.2.1 Functional categories

Headless phrases are only permitted for functional categories.

(219)
$$\begin{array}{llcc} \text{CP} & \to & \text{XP} & \text{C}' \\ & & \Big\{(\uparrow \text{SUBJ})|(\uparrow \text{DF})\Big\} = \downarrow & \uparrow=\downarrow \end{array}$$

$$\text{(220)}\quad \mathrm{C}' \;\rightarrow\; \begin{array}{cc} \mathrm{C} & \{\,\mathrm{IP} \mid \mathrm{S} \mid \mathrm{VP}\,\} \\ \uparrow=\downarrow & \uparrow=\downarrow \end{array}$$

$$\text{(221)}\quad \mathrm{DP} \;\rightarrow\; \begin{array}{cc} \mathrm{DP} & \mathrm{D}' \\ (\uparrow \textsc{poss}) = \downarrow & \uparrow=\downarrow \\ (\uparrow \textsc{def}) = + & \\ (\downarrow \textsc{case}) =_c \textsc{gen} & \end{array}$$

$$\text{(222)}\quad \mathrm{D}' \;\rightarrow\; \begin{array}{cc} \mathrm{D} & \mathrm{NP} \\ \uparrow=\downarrow & \uparrow=\downarrow \end{array}$$

$$\text{(223)}\quad \mathrm{IP} \;\rightarrow\; \begin{array}{cc} \Big\{\, \mathrm{DP} \mid \mathrm{NP} \mid \mathrm{PP} \mid \mathrm{CP} \,\Big\} & \mathrm{I}' \\ (\uparrow \textsc{subj}) = \downarrow & \uparrow=\downarrow \\ \Big((\uparrow \textsc{df}) = \downarrow \Big) & \end{array}$$

$$\text{(224)}\quad \mathrm{I}' \;\rightarrow\; \begin{array}{cc} \mathrm{I} & \mathrm{VP} \\ \uparrow=\downarrow & \uparrow=\downarrow \end{array}$$

A.1.2.2 Exocentric category S

$$\text{(225)}\quad \mathrm{S} \;\rightarrow\; \begin{array}{cc} \Big\{\, \mathrm{DP} \mid \mathrm{NP} \mid \mathrm{PP} \mid \mathrm{CP} \,\Big\} & \mathrm{VP} \\ (\uparrow \textsc{subj}) = \downarrow & \uparrow=\downarrow \\ \Big((\uparrow \textsc{df}) = \downarrow \Big) & \end{array}$$

A.1.2.3 Lexical categories

$$\text{(226)}\quad \mathrm{AP} \;\rightarrow\; \begin{array}{ccc} \mathrm{A} & \mathrm{PP}^{*} & \{\mathrm{CP} \mid \mathrm{IP} \mid \mathrm{S}\} \\ \uparrow=\downarrow & (\uparrow (\downarrow \textsc{pcase})) = \downarrow & (\uparrow \textsc{comp}) = \downarrow \end{array}$$

$$\text{(227)}\quad \mathrm{NP} \;\rightarrow\; \begin{array}{ccc} \mathrm{N} & \mathrm{PP}^{*} & \mathrm{CP} \\ \uparrow=\downarrow & (\uparrow (\downarrow \textsc{pcase})) = \downarrow & (\uparrow \textsc{comp}) = \downarrow \end{array}$$

$$\text{(228)}\quad \mathrm{PP} \;\rightarrow\; \begin{array}{cccc} \mathrm{P} & \{\mathrm{DP}|\mathrm{NP}\} & \mathrm{PP} & \{\mathrm{IP}|\mathrm{S}\} \\ \uparrow=\downarrow & (\uparrow \textsc{obj}) =\downarrow & (\uparrow (\downarrow \textsc{pcase})) = \downarrow & (\uparrow \textsc{comp}) = \downarrow \end{array}$$

$$\text{(229)}\quad \mathrm{VP} \;\rightarrow\; \begin{array}{ccccc} \mathrm{V} & \{\mathrm{DP}|\mathrm{NP}\} & \mathrm{XP} & \mathrm{PP}^{*} & \{\mathrm{CP}|\mathrm{IP}|\mathrm{S}\} \\ \uparrow=\downarrow & (\uparrow [+o]) =\downarrow & (\uparrow \textsc{xcomp}) = \downarrow & (\uparrow (\downarrow \textsc{pcase})) = \downarrow & (\uparrow \textsc{comp}) = \downarrow \\ & & \neg(\downarrow \textsc{tense}) & & \end{array}$$

A.1.2.4 Adjunction

$$\text{(230)}\quad \mathrm{Z} \;\longrightarrow\; \begin{array}{cc} \mathrm{XP} & \mathrm{Z} \\ (\uparrow \textsc{df}) = \downarrow & \uparrow=\downarrow \\ (\downarrow \textsc{gf}^{*} \textsc{pron}) \neq \textsc{wh} & \end{array}$$

$$\text{(231)}\quad \mathrm{AP} \;\longrightarrow\; \begin{array}{ccc} \mathrm{ADVP}^{*} & \mathrm{AP} & \mathrm{PP}^{*} \\ \downarrow \in (\uparrow \textsc{adj}) & \uparrow=\downarrow & \downarrow \in (\uparrow \textsc{adj}) \end{array}$$

$$(232)\quad \text{NP} \longrightarrow \begin{array}{c}\text{ADJP*}\\ \downarrow \in (\uparrow\ \textsc{adj})\end{array} \quad \begin{array}{c}\text{NP}\\ \uparrow=\downarrow\end{array} \quad \begin{array}{c}\text{PP*}\\ \downarrow \in (\uparrow\ \textsc{adj})\end{array} \quad \begin{array}{c}\Big\{\text{CP} \mid \text{IP} \mid \text{S}\Big\}^{*}\\ \downarrow \in (\uparrow\ \textsc{adj})\\ \Big((\downarrow \textsc{oper pred}) = \text{‘}\textsc{pro}\text{’}\Big)\end{array}$$

$$(233)\quad \text{PP} \longrightarrow \begin{array}{c}\text{ADVP*}\\ \downarrow \in (\uparrow\ \textsc{adj})\end{array} \quad \begin{array}{c}\text{PP}\\ \uparrow=\downarrow\end{array} \quad \begin{array}{c}\text{PP*}\\ \downarrow \in (\uparrow\ \textsc{adj})\end{array}$$

$$(234)\quad \text{VP} \longrightarrow \begin{array}{c}\Big\{\text{ADVP} \mid \text{PP}\Big\}^{*}\\ \downarrow \in (\uparrow\ \textsc{adj})\end{array} \quad \begin{array}{c}\text{VP}\\ \uparrow=\downarrow\end{array} \quad \begin{array}{c}\Big\{\text{ADVP}\mid \text{PP}\Big\}^{*}\\ \downarrow \in (\uparrow\ \textsc{adj})\end{array}$$

A.2 Korean

The phrase-structure rules below, used in the thesis, are derived from the proposal of Cho and Sells (1995). Categories in the grammar are N and V; the maximal projection in the grammar is X$'$.

The variable $\widehat{\text{W}}$ stands for any non-projecting category.

$$(235)\quad \text{N}' \rightarrow \left\{ \begin{array}{c}\text{N}\\ (\hat{*}\ \textsc{type}) = \textsc{type}\\ \uparrow=\downarrow\end{array} \;\middle|\; \left\{ \begin{array}{c}\text{N}'\\ \textsc{type:n-sis}\\ (\uparrow \textsc{gf}) =\downarrow\end{array} \;\middle|\; \begin{array}{c}\text{V}'\\ \textsc{type:n-sis}\\ (\uparrow \textsc{gf}) =\downarrow\end{array} \right\} \begin{array}{c}\text{N}'\\ (\hat{*}\ \textsc{type}) = \textsc{type}\\ \uparrow=\downarrow\end{array} \right\}$$

$$(236)\quad \text{V}' \rightarrow \left\{ \begin{array}{c}\text{V}\\ (\hat{*}\ \textsc{type}) = \textsc{type}\\ \uparrow=\downarrow\end{array} \;\middle|\; \left\{ \begin{array}{c}\text{N}'\\ \textsc{type:v-sis}\\ (\uparrow \textsc{gf}) =\downarrow\end{array} \;\middle|\; \begin{array}{c}\text{V}'\\ \textsc{type:v-sis}\\ (\uparrow \textsc{gf}) =\downarrow\end{array} \right\} \begin{array}{c}\text{V}'\\ (\hat{*}\ \textsc{type}) = \textsc{type}\\ \uparrow=\downarrow\end{array} \right\}$$

$$(237)\quad \text{N} \rightarrow \begin{array}{c}\widehat{\text{W}}\\ \textsc{type:n-sis}\\ (\uparrow \textsc{gf}) = \downarrow\end{array} \quad \begin{array}{c}\text{N}\\ (\hat{*}\ \textsc{type}) = \textsc{type}\\ \uparrow=\downarrow\end{array}$$

$$(238)\quad \text{V} \rightarrow \begin{array}{c}\widehat{\text{W}}\\ \textsc{type:v-sis}\\ (\uparrow \textsc{gf}) = \downarrow\end{array} \quad \begin{array}{c}\text{V}\\ (\hat{*}\ \textsc{type}) = \textsc{type}\\ \uparrow=\downarrow\end{array}$$

Bibliography

Aho, A. V. and Ullman, J. D. (1972). *The theory of parsing, translation and compiling: Volume 1. Parsing.* Prentice Hall, Englewood Cliffs, NJ.

Ajdukiewicz, K. (1935). Die syntaktische konnexität. *Studia Philosophica*, 1:1–27.

Anderson, J. R. (1990). *The adaptive character of thought.* Lawrence Erlbaum Associates, Hillsdale, N.J.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press, New York.

Anderson, J. R., Budiu, R., and Reder, L. M. (2001). A theory of sentence memory as part of a general theory of memory. *Journal of Memory and Language*, 45(3):337–367.

Asudeh, A. (2012). *The logic of pronominal resumption.* Number 35 in Oxford Studies in Theoretical Linguistics. Oxford University Press, Oxford.

Asudeh, A. and Giorgolo, G. (2012). Flexible composition for optional and derived arguments. In Butt, M. and King, T. H., editors, *Proceedings of LFG12*, pages 64–84, Stanford, CA. CSLI Publications.

Bailey, K. G. and Ferreira, F. (2003). Disfluencies affect the parsing of garden-path sentences. *Journal of Memory and Language*, 49:183–200.

Ball, J. T. (2003). Double-R grammar. `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.1494&rep=rep1&type=pdf`.

Ball, J. T. (2004). A cognitively plausible model of language comprehension. In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation*, pages 305–316.

Ball, J. T. (2005). A bi-polar theory of nominal and clause structure. In *Proceedings of the Cognitive Science Society*, volume 27, pages 190–195.

Ball, J. T. (2007a). A bi-polar theory of nominal and clause structure and function. *Annual Review of Cognitive Linguistics*, 5(1):27–54.

Ball, J. T. (2007b). Construction-driven language processing. In *Proceedings of the 2nd European Cognitive Science Conference*, pages 722–727.

Ball, J. T. (2010). Projecting grammatical features in nominals: Cognitive processing theory & computational implementation. In *Proceedings of the 19th Conference on Behavior Representation in Modeling and Simulation, Charleston, SC, 21 - 24 March 2010*, pages 55–62.

Ball, J. T. (2011a). Explorations in ACT-R based cognitive modeling — chunks, inheritance, production matching and memory in language analysis. In *AAAI Fall Symposium: Advances in Cognitive Systems*, pages 10–17.

Ball, J. T. (2011b). A pseudo-deterministic model of human language processing. In *Proceedings of the Cognitive Science Society*, volume 33, pages 495–500.

Ball, J. T. (2012a). Explorations in ACT-R based language analysis – memory chunk activation, retrieval and verification without inhibition. In *Proceedings of the 11th International Conference on Cognitive Modeling*, pages 131–136.

Ball, J. T. (2012b). The representation and processing of tense, aspect & voice across verbal elements in English. In *Proceedings of the Cognitive Science Society*, volume 34, pages 1308–1313.

Ball, J. T. (2013). Modeling the binding of implicit arguments in complement clauses in ACT-R/Double-R. In *Proceedings of the 12th International Conference on Cognitive Modeling*, pages 84–88.

Ball, J. T., Heiberg, A., and Silber, R. (2007). Toward a large-scale model of language comprehension in ACT-R 6. In *Proceedings of the 8th international conference on cognitive modeling*, pages 173–179.

Bank, S. (2013). Graphviz. https://www.graphviz.org.

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58.

Barton, S. B. and Sanford, A. J. (1993). A case study of anomaly detection: Shallow semantic processing and cohesion establishment. *Memory & Cognition*, 21(4):477–487.

Baudiffier, V., Caplan, D., Gaonac'h, D., and Chesnet, D. (2011). The effect of noun animacy on the processing of unambiguous sentences: evidence from French relative clauses. *The Quarterly Journal of Experimental Psychology*, 64:1896–1905.

Bergen, B. and Chang, N. (2013). Embodied Construction Grammar. In Hoffman, T. and Trousdale, G., editors, *The Oxford Handbook of Construction Grammar*, chapter 10, pages 168–190. Oxford University Press, New York.

Betancort, M., Carreiras, M., and Sturt, P. (2009). The processing of subject and object relative clauses in Spanish: an eye-tracking study. *The Quarterly Journal of Experimental Psychology*, 62:1915–1929.

Bever, T. G. (1970). The cognitive basis for linguistic structures. In Hayes, R., editor, *Cognition and language development*, pages 279–362. Wiley & Sons, New York.

Block, H.-U. and Hunze, R. (1986). Incremental construction of c- and f-structure in an LFG-parser. In *Coling 1986 Volume 1: The 11th International Conference on Computational Linguistics*, pages 490–493.

Boas, H. C. and Sag, I. A. (2012). *Sign-based construction grammar*. CSLI Publications/Center for the Study of Language and Information.

Bögel, T., Butt, M., Kaplan, R. M., King, T. H., and Maxwell, J. T. (2009). Prosodic phonology in LFG: A new proposal. In Butt, M. and King, T. H., editors, *Proceedings of the LFG09 Conference*, pages 146–166.

Bornkessel, I. and Schlesewsky, M. (2006). The extended argument dependency model: A neurocognitive approach to sentence comprehension across languages. *Psychological Review*, 113(4):787–821.

Bornkessel-Schlesewsky, I. and Schlesewsky, M. (2008). An alternative perspective on "semantic P600" effects in language comprehension. *Brain Research REviews*, 59:55–73.

Bornkessel-Schlesewsky, I. and Schlesewsky, M. (2013). Reconciling time, space and function: A new dorsal-ventral stream model of sentence comprehension. *Brain and Language*, 125:60–76.

Bothell, D. (2017). ACT-R 7 Reference Manual. Technical report, Carnegie Mellon University, Pittsburgh, PA.

Brasoveanu, A. and Dotlačil, J. (2018). A cognitively realistic left-corner parser with visual and motor interfaces. In *Proceedings of the 2018 CUNY conference on human sentence processing.*

Bresnan, J. (1995). Linear order, syntactic rank, and empty categories: On weak crossover. In Dalrymple, M., Kaplan, R. M., Maxwell, J. T., and Zaenen, A. E., editors, *Formal Issues in Lexical-Functional Grammar*, pages 241–274. CSLI Publications.

Bresnan, J. (2001). *Lexical-functional syntax*, volume 16 of *Blackwell textbooks in linguistics*. Blackwell, Oxford.

Bresnan, J., Asudeh, A., Toivonen, I., and Wechsler, S. (2016). *Lexical-functional syntax*. Wiley Blackwell, Chichester, West Sussex, 2nd edition.

Bresnan, J. and Kanerva, J. M. (1989). Locative inversion in Chicheŵa: A case study of factorization in grammar. *Linguistic Inquiry*, 20(1):1–50.

Bresnan, J., Kaplan, R. M., Peters, S., and Zaenen, A. (1982). Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 13(4):613–635.

Bresnan, J. and Mchombo, S. A. (1987). Topic, pronoun, and agreement in Chicheŵa. *Language*, 63(4):741–782.

Brouwer, H., Fitz, H., and Hoeks, J. (2012). Getting real about Semantic Illusions: Rethinking the functional role of the P600 in language comprehension. *Brain Research*, 1446:127–143.

Budiu, R. and Anderson, J. R. (2004). Interpretation-based processing: a unified theory of semantic sentence comprehension. *Cognitive Science*, 28:1–44.

Butt, M. (1995). *The structure of complex predicates in Urdu*. CSLI Publications, Stanford, CA.

Cann, R., Kempson, R., and Marten, L. (2005). *The dynamics of language*. Elsevier.

Carlson, K., Clifton, Jr., C., and Frazier, L. (2001). Prosodic boundaries in adjunct attachment. *Journal of Memory and Language*, 45:58–81.

Cho, Y.-M. Y. and Sells, P. (1995). A lexical account of inflectional suffixes in Korean. *Journal of East Asian Linguistics*, 4(2):119–174.

Chomsky, N. (1965). *Aspects of the theory of syntax*. MIT Press, Cambridge, MA.

Chomsky, N. (1968). *Remarks on nominalization*. Linguistics Club, Indiana University.

Chomsky, N. (1981). *Lectures on government and binding*. Foris, Dordrecht.

Chomsky, N. (1986). *Barriers*. MIT Press, Cambridge, MA.

Chomsky, N. (1995). *The minimalist program*, volume 28. Cambridge Univ Press.

Chow, W.-Y., Smith, C., Lau, E., and Phillips, C. (2015). A “bag-of-arguments” mechanism for initial verb predictions. *Language, Cognition and Neuroscience*, page DOI: 10.1080/23273798.2015.1066832.

Christianson, K., Hollingworth, A., Halliwell, J., and Ferreira, F. (2001). Thematic roles assigned along the garden path linger. *Cognitive Psychology*, 42:368–407.

Clifton, Jr., C., Carlson, K., and Frazier, L. (2002). Informative prosodic boundaries. *Language and Speech*, 45(2):87–114.

Clifton, Jr., C. and Frazier, L. (1989). Comprehending sentences with long-distance dependencies. In Carlson, G. N. and Tanenhaus, M. K., editors, *Linguistic structure in language processing*, pages 273–317. Kluwer Academic Publishers.

Crouch, D., Dalrymple, M., Kaplan, R. M., King, T. H., Maxwell III, J. T., and Newman, P. S. (2008). *XLE Documentation*. Xerox Palo Alto Research Center, Palo Alto, CA.

Cuetos, F. and Mitchell, D. C. (1988). Cross-linguistic differences in parsing: Restrictions on the use of the Late Closure strategy in Spanish. *Cognition*, pages 73–105.

Culicover, P. and Jackendoff, R. (2005). *Simpler syntax.* Oxford University Press, Oxford.

Dalrymple, M. (1993). *The syntax of anaphoric binding.* CSLI Publications, Stanford, CA.

Dalrymple, M., editor (1999). *Semantics and syntax in Lexical Functional Grammar.* MIT Press.

Dalrymple, M. (2001). *Lexical Functional Grammar.* Academic Press, San Diego.

Dalrymple, M. (2017). Unlike phrase structure category co-ordination. *Bergen Language and Linguistic Studies*, 8:33–55.

Dalrymple, M., Kaplan, R. M., and King, T. H. (2001). Weak crossover and the absence of traces. In Butt, M. and King, T. H., editors, *Proceedings of the LFG01 Conference*, pages 66–82. CSLI Publications.

Dalrymple, M. and King, T. H. (2013). Nested and crossed dependencies and the existence of traces. In King, T. H. and de Paiva, V., editors, *From Quirky Case to Representing Space: Papers in Honor of Annie Zaenen*, chapter 11, pages 139–151. CSLI Publications.

Dalrymple, M. and Mycock, L. (2011). The prosody-semantics interface. In Butt, M. and King, T. H., editors, *Proceedings of LFG11*, Stanford, CA. CSLI Publications.

Dillon, B., Mishler, A., Sloggett, S., and Phillips, C. (2013). Contrasting intrusion profiles for agreement and anaphora: Experimental and modeling evidence. *Journal of Memory and Language*, 69:85–103.

Duffy, S. A., Morris, R. K., and Rayner, K. (1988). Lexical ambiguity and fixation times in reading. *Journal of Memory and Language*, 27:429–446.

Durrell, M. (2002). *Hammer's German grammar and usage.* Arnold, London, 4th edition.

Dutta, S. and Basu, A. (2012). A cognitive approach to word sense disambiguation. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 7181 of *CICLing 2012. Lecture Notes in Computer Science.* Springer, Berlin, Heidelberg.

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

Eliasmith, C. (2013). *How to build a brain: a neural architecture for biological cognition.* Oxford University Press, New York.

Emond, B. (1999). Estimating processing time of online semantic interpretation components. In *Proceedings of the Fifth Annual ACT-R Workshop*, pages 113–118, Fairfax, VA. Department of Psychology, George Mason University.

Engelmann, F. (2015). Inter-ACT: Simple R simulation of ACT-R retrieval process. `https://github.com/felixengelmann/inter-act`.

Engelmann, F. (2016). An integrated model of eye movement behaviour in sentence comprehension. `https://github.com/felixengelmann/act-r-sentence-parser-em`.

Erickson, T. D. and Mattson, M. E. (1981). From words to meaning: A semantic illusion. *Journal of Verbal Learning and Verbal Behaviour*, 20:541–551.

Falk, Y. N. (2001). *Lexical-Functional Grammar: An introduction to parallel constraint-based syntax.* CSLI Publications, Stanford, CA.

Falk, Y. N. (2005). Open argument functions. In Butt, M. and King, T. H., editors, *Proceedings of the LFG05 Conference.* CSLI Publications.

Ferreira, F. (2005). Psycholinguistics, formal grammars and cognitive science. In *The Linguistic Review*, volume 22, pages 365–380. Walter de Gruyter.

Ferreira, F., Bailey, K. G., and Ferraro, V. (2002). Good-enough representations in language comprehension. *Current Directions in Psychological Science*, pages 11–15.

Ferreira, F. and Henderson, J. M. (1991). Recovery from misanalyses of garden-path sentences. *Journal of Memory and Language*, 30:725–745.

Ferreira, F. and Patson, N. D. (2007). The 'Good Enough' approach to language comprehension. *Language and Linguistics Compass*, 1/1-2:71–83.

Findlay, J. Y. (2016). Mapping theory without argument structure. *Journal of Language Modelling*, 4(2):293–338.

Fodor, J. A. and Garrett, M. (1967). Some syntactic determinants of sentential complexity. *Perception and Psychophysics*, 2(7):289–296.

Fodor, J. D. (1978). Parsing strategie and constraints on transformations. *Linguistic Inquiry*, 9(3):427–473.

Fodor, J. D. (2002). Psycholinguistics cannot escape prosody. In *Speech Prosody 2002, Aix-en-Provence*. International Speech Communiation Association.

Fodor, J. D. (2013). Pronouncing and comprehending center-embedded sentences. In Sanz, M., Laka, I., and Tanenhaus, M. K., editors, *Language down the garden path: The cognitive and biological basis for linguistic structures*, chapter 9, pages 206–228. Oxford University Press, Oxford.

Fodor, J. D. and Ferreira, F., editors (1998). *Reanalysis in sentence processing*. Kluwer, Dordrecht.

Ford, M., Bresnan, J., and Kaplan, R. M. (1982). A competence-based theory of syntactic closure. In Bresnan, J., editor, *The mental representation of grammatical relations*, chapter 11, pages 727–796. MIT Press, Cambridge, MA.

Frank, A. (2003). Projecting LFG f-structures from chunks – or (Non-)configurationality from a different viewpoint. In Butt, M. and King, T. H., editors, *Proceedings of the LFG03 Conference*, pages 217–237.

Frazier, L. (1978). *On comprehending sentences: Syntactic parsing strategies*. PhD thesis, University of Connecticut.

Frazier, L. (1987). Syntactic processing: evidence from Dutch. *Natural Language and Linguistic Theory*, 5:519–559.

Frazier, L. and Clifton, Jr., C. (1996). *Construal*. The MIT Press, Cambridge, MA.

Frazier, L. and Fodor, J. D. (1978). The sausage machine: A new two-stage parsing model. *Cognition*, 6:291–325.

Freiman, M. and Ball, J. T. (2010). Improving the reading rate of Double-R-language. In *Proceedings of the 10th International Conference on Cognitive Modeling*, pages 67–72.

Friederici, A. D. (2011). The brain basis of language processing: From structure to function. *Physiological Review*, 91:1357–1392.

Friederici, A. D. (2012). The cortical language circuit: from auditory perception to sentence comprehension. *Trends in Cognitive Sciences*, 16(5):262–268.

Gansner, E. R., Koutsofios, E., and North, S. (2015). Draw graphs with *dot*. https://www.graphviz.org/pdf/dotguide.pdf.

Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Gibson, E. (2000). The Dependency Locality Theory: A distance-based theory of linguistic complexity. In Marantz, A., Miyashita, Y., and O'Neil, W. A., editors, *Image, language, brain: papers from the First Mind Articulation Project Symposium*, chapter 5, pages 95–126. MIT Press, Cambridge, MA.

Gibson, E. and Hickok, G. (1993). Sentence processing with empty categories. *Language and Cognitive Processes*, 8(2):147–161.

Gibson, E., Pearlmutter, N. J., Canseco-Gonzales, E., and Hickok, G. (1996). Recency preference in the human sentence processing mechanism. *Cognition*, 59:23–59.

Gibson, E. and Wu, H.-H. I. (2013). Processing Chinese relative clauses in context. *Language and Cognitive Processes*, 28(1/2):125–155.

Gouvea, A. C., Phillips, C., Kazanina, N., and Poeppel, D. (2010). The linguistic processes underlying the P600. *Language and Cognitive Processes*, 25(2):149–188.

Grimshaw, J. (2000). Locality and extended projection. In Coopmans, P., Everaert, M., and Grimshaw, J., editors, *Lexical specification and insertion*, pages 115–134. John Benjamins, Amsterdam.

Grodner, D. and Gibson, E. (2005). Consequences of the serial nature of linguistic input for sentential complexity. *Cognitive science*, 29(2):261–290.

Grodner, D., Gibson, E., and Tunstall, S. (2002). Syntactic complexity in ambiguity resolution. *Journal of Memory and Language*, 46(2):267–295.

Gruber, J. S. (1962). *Studies in lexical relations.* PhD thesis, Massachusetts Institute of Techonology.

Hagoort, P. (2005). On Broca, brain, and binding: a new framework. *Trends in Cognitive Sciences*, 9(9):416–423.

Hagoort, P., Baggio, G., and Willems, R. M. (2009). Semantic unification. In Gazzaniga, M. S., editor, *The cognitive neurosciences, 4th edition*, chapter 56, pages 819–836. MIT Press.

Hagoort, P. and Indefrey, P. (2014). The neurobiology of language beyond single words. *Annual Review of Neuroscience*, 37:347–362.

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of NAACL*, volume 2, pages 159–166.

Hale, J. (2016). Information-theoretical complexity metrics. *Language and Linguistics Compass*, 10(9):397–412.

Hale, J. T. (2014). *Automaton theories of human sentence comprehension.* CSLI Publications, Stanford, CA.

Halliday, M. (1967). Notes on transitivity and theme in English part 2. *Journal of Linguistics*, 3:177–274.

Han, C.-H. (2013). On the syntax of relative clauses in Korean. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 58(2):319–347.

Haug, D. T. T. (2014). Partial Dynamic Semantics for anaphora: Compositionality without syntactic coindexation. *Journal of Semantics*, 31:457–511.

Heiberg, A., Harris, J., and Ball, J. T. (2007). Dynamic visualization of ACT-R declarative memory structure. In *Proceedings of the 8th International Conference on Cognitive Modeling*, pages 233–234.

Hendriks, P., van Rijn, H., and Valkenier, B. (2007). Learning to reason about speakers' alternatives in sentence comprehension: A computational account. *Lingua*, 117:1879–1896.

Hickok, G. (2009). The functional neuroanatomy of language. *Physics of Life Reviews*, 6:121–143.

Hickok, G. and Poeppel, D. (2004). Dorsal and ventral streams: a framework for understanding aspects of the functional anatomy of language. *Cognition*, 92:67–99.

Holmes, V. M. and O'Regan, J. (1981). Eye fixation patterns during the reading of relative-clause sentences. *Journal of Verbal Learning and Verbal Behaviour*, 20(417-430).

Hwang, H. and Schafer, A. J. (2009). Constituent length affects prosody and processing for a Dative NP ambiguity in Korean. *Journal of Psycholinguistic Research*, 38:151–175.

Hwang, H. and Steinhauer, K. (2011). Phrase length matters: The interplay between implicit prosody and syntax in Korean "garden path" sentences. *Journal of Cognitive Neuroscience*, 23(11):3555–3575.

Ihm, H. B., Hong, K. P., and Chang, S. I. (2001). *Korean grammar for international learners.* Yonsei University Press, Seoul.

Indefrey, P. (2011). The spatial and temporal signatures of word production components: a critical update. *Frontiers in Psychology*, 2(255):1–16.

Jackendoff, R. (2011). What is the human language faculty?: Two views. *Language*, 87(3):586–624.

Jackendoff, R. S. (1977). *X Syntax: A study of phrase structure.* MIT Press, Cambridge, MA.

Jäger, L. A., Engelmann, F., and Vasishth, S. (2015). Retrieval interference in reflexive processing: experimental evidence from Mandarin, and computational modeling. *Frontiers in Psychology*, 6(617):1–24.

Jäger, L. A., Engelmann, F., and Vasishth, S. (2017). Similarity-based interference in sentence comprehension: Literature review and Bayesian meta-analysis. *Journal of Memory and Language*, 94:316–339.

Jäger, L. A., Mertzen, D., Van Dyke, J. A., and Vasishth, S. (2019). Interference patterns in subject-verb agreement and reflexives revisited: A large-sample study. [Pre-publication online at https://psyarxiv.com/7c4gu. Under revision].

Janda, L. and Townsend, C. E. (2000). *Czech.* Lincom Europa, Munich.

Jones, S. (2017). Developing a computational model of language processing using LFG. Poster presented at the LFG17 Conference, Universität Konstanz, Germany.

Jun, S.-A. (2005a). Korean intonational phonology and prosodic transcription. In *Prosodic typology: the phonology of intonation and phrasing*, chapter 8, pages 202–228. Oxford, UK: Oxford University Press.

Jun, S.-A., editor (2005b). *Prosodic typology: the phonology of intonation and phrasing.* Oxford University Press, Oxford.

Kaplan, R. M. (1972). Automated Transition Networks as psychological models of human sentence comprehension. *Artificial Intelligence*, 3:77–100.

Kaplan, R. M. (1995). The formal architecture of Lexical-Functional Grammar. In Dalrymple, M., Kaplan, R. M., Maxwell, J. T., and Zaenen, A. E., editors, *Formal Issues in Lexical-Functional Grammar.* CSLI Publications, Stanford, CA.

Kaplan, R. M. and Bresnan, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In Bresnan, J., editor, *The mental representation of grammatical relations.* MIT Press.

Kaplan, R. M. and Maxwell, John T., I. (1996). Grammar writer's workbench. Technical Report Version 3.1, Xerox Palo Alto Research Center, Palo Alto, CA. [Online: accessed 2018-09-11].

Karimi, H. and Ferreira, F. (2015). Good-enough linguistic representations and online cognitive equilibrium in language processing. *Quarterly Journal of Experimental Psychology.*

Kazanina, N., Lau, E. F., Lieberman, M., Yoshida, M., and Phillips, C. (2007). The effect of syntactic constraints on the processing of backwards anaphora. *Journal of Memory and Language*, 56:384–409.

Keenan, E. L. and Comrie, B. (1977). Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8(1):63–99.

Kempson, R. and Kiaer, J. (2010). Multiple long-distance scrambling: Syntax as reflections of processing. *Journal of Linguistics*, 46:127–192.

Kiaer, J. (2011). Word-order variation in Korean and the architecture of grammar. In Kempson, R., Gregoromichelaki, E., and Howes, C., editors, *The dynamics of lexical interfaces*, chapter 6, pages 165–203. CSLI Publications, Stanford, CA.

Kim, A. and Osterhout, L. (2005). The independence of combinatory semantic processing: Evidence from event-related potentials. *Journal of Memory and Language*, 52:205–225.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47.

King, J. and Just, M. A. (1991). Individual differences in syntactic processing: the role of working memory. *Journal of Memory and Language*, 30:580–602.

King, T. H. and Zaenen, A. (2004). F-structures, information structure, and discourse structure. http://web.stanford.edu/group/cslipublications/cslipublications/LFG/9/abstracts/lfg04kingzaenen-abs.html.

Kjelgaard, M. M. and Speer, S. R. (1999). Prosodic facilitaion and interface in the resolution of temporary syntactic closure ambiguity. *Journal of Memory and Language*, 40:153–194.

Kos, M., Vosse, T., van den Brink, D., and Hagoort, P. (2010). About edible restaurants: conflicts between syntax and semantics as revealed by ERPs. *Frontiers in Psychology*, 1(222):1–11.

Kuperberg, G. R. (2007). Neural mechanisms of language comprehension: Challenges to syntax. *Brain Research*, pages 23–49.

Kuperberg, G. R. and Jaeger, T. F. (2016). What do we mean by prediction in language comprehension? *Language, Cognition and Neuroscience*, 31(1):32–59.

Kush, D. and Phillips, C. (2014). Local anaphor licensing in an SOV language: implications for retrieval strategies. *Frontiers in Psychology*, 5(1252):1–12.

Kutas, M. and Federmeier, K. D. (2011). Thirty years and counting: Finding meaning in the N400 component of the event related brain potential (ERP). *Annual Review of Psychology*, 62:621–647.

Kutas, M. and Hillyard, S. A. (1980). Reading between the lines: Event-related brain potentials during natural sentence processing. *Brain and Language*, 11:354–373.

Kutas, M. and Hillyard, S. A. (1984). Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307:161–163.

Kwon, N., Gordon, P. C., Lee, Y., Kluender, R., and Polinsky, M. (2010). Cognitive and linguistic factors affective subject/object asymmetry: An eye-tracking study of prenominal relative clauses in Korean. *Language*, 86(3):546–582.

Kwon, N., Kluender, R., Kutas, M., and Polinsky, M. (2013). Subject/object processing asymmetries in Korean relative clauses: Evidence from ERP data. *Language*, 89(3):537–585.

Laird, J. E. (2012). *The Soar Cognitive Architecture.* MIT Press, Cambridge, MA.

Laird, J. E., Lebiere, C., and Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, 38(4):13–26.

Langacker, R. W. (1986). An introduction to Cognitive Grammar. *Cognitive Science*, 10(1):1–40.

Langacker, R. W. (1987). *Foundations of Cognitive Grammar, Volume 1.* Stanford University Press, Stanford, CA.

Langacker, R. W. (1991). *Foundations of Cognitive Grammar, Volume 2: Descriptive applications.* Stanford University Press, Stanford, CA.

Langus, A., Marchetto, E., Bion, R. A. H., and Nespor, M. (2012). Can prosody be used to discover hierarchical structure in continuous speech? *Journal of Memory and Language*, 66:285–306.

Levelt, W. J. M. (1989). *Speaking: From intention to articulation.* MIT Press.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Lewis, R. L. (1999). *Attachment without competition: A race-based model of ambiguity resolution in a limited working memory.* Presented at the CUNY Sentence Processing Conference, New York. http://act-r.psy.cmu.edu/wordpress/wp-content/uploads/2013/09/252lewis_99.pdf.

Lewis, R. L. and Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29:375–419.

Lindes, P. (2018). The Common Model of Cognition and humanlike language comprehension. *Procedia Computer Science*, 145:765–772.

Lindes, P. and Laird, J. E. (2016). Toward integrating cognitive linguistics and cognitive language processing. In *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016)*, University Park, PA. Penn State.

Lindes, P. and Laird, J. E. (2017). Cognitive modeling approaches to language comprehension using Construction Grammar. Technical Report SS-17-02, Association for the Advancement of Artificial Intelligence.

Lowe, J. J. (2016). English possessive *'s*: clitic and affix. *Natural Language and Linguistic Theory*, 34:157–195.

Lupyan, G. and Clark, A. (2015). Words and the world: Predictive coding and the Language-Perception-Cognition interface. *Current Directions in Psychological Science*, 24(4):279–284.

MacDonald, M. C., Pearlmutter, N. J., and Seidenberg, M. S. (1994). Lexical nature of syntactic ambiguity resolution. *Psychological Review*, 101(4):676–703.

Mak, W. M., Vonk, W., and Schriefers, H. (2002). The influence of animacy on relative clause processing. *Journal of Memory and Language*, 47(50-68).

Marslen-Wilson, W., Tyler, L., Warren, P., Grenier, P., and Lee, C. (1992). Prosodic effects in minimal attachment. *The Quarterly Journal of Experimental Psychology*, 45A(1):73–87.

McElree, B. and Dyer, L. (2013). Beyond capacity: The role of memory processes in building linguistic structure in real time. In Sanz, M., Laka, I., and Tanenhaus, M. K., editors, *Language down the garden path: The cognitive and biological basis for linguistic structures*, chapter 10, pages 229–240. Oxford University Press.

McElree, B., Foraker, S., and Dyer, L. (2003). Memory structures that subserve sentence comprehension. *Journal of Memory and Language*, 48(1):67–91.

Misker, J. M. and Anderson, J. R. (2003). Combining Optimality Theory and a cognitive architecture. In *Proceedings of the 5th international conference on cognitive modeling*, Bamberg, Germany.

Miyamoto, E. T. and Nakamura, M. (2003). Subject/object asymmetries in the processing of relative clauses in Japanese. In Garding, G. and Tsujimura, M., editors, *WCCFL 22 Proceedings*, pages 342–355, Somerville, MA. Cascadilla Press.

Mohan, S., Mininger, A. H., and Laird, J. E. (2013). Towards an indexical model of situated language comprehension for real-world cognitive agents. *Advances in Cognitive Systems*, 3:163–182.

Murphy, B., Wehbe, L., and Fyshe, A. (2018). Decoding language from the brain. In Poibeau, T. and Villavicencio, A., editors, *Language, cognition, and computational models*, chapter 3, pages 53–80. Cambridge University Press.

Muskens, R. (1996). Combining Montague semantics and Discourse Representation. *Linguistics and Philosophy*, 19:143–186.

Mycock, L. and Lowe, J. (2013). The prosodic marking of discourse functions. In Butt, M. and King, T. H., editors, *Proceedings of LFG13*, Stanford, CA. CSLI Publications.

Nadathur, P. (2013). Weak crossover and the direct association hypothesis. In Butt, M. and King, T. H., editors, *Proceedings of the LFG13 Conference*. CSLI Publications.

Neville, H., Nicol, J. L., Barss, A., Forster, K. I., and Garrett, M. F. (1991). Syntactically based sentence processing classes: Evidence from event-related brain potentials. *Journal of Cognitive Neuroscience*, 3(2):151–165.

Newell, A., Yost, G., Laird, J. E., Rosenbloom, P. S., and Altmann, E. (1991). Formulating the problem space computational model. In Rashid, R., editor, *Carnegie Mellon Computer Science: A 25 Year Commemorative*. ACM Press/Addison/Wesley.

Nordlinger, R. (1998). *Constructive case: Evidence from Australian languages.* CSLI Publications, Stanford, CA.

Parker, D., Shvartsman, M., and Van Dyke, J. A. (2017). The cue-based retrieval theory of sentence comprehension: New findings and new challenges. In Escobar, L., Torrens, V., and Parodi, T., editors, *Language processing and disorders*, pages 121–144. Cambridge Scholars Publishing, Newcastle upon Tyne.

Parker, D. J. (2014). *The cognitive basis for encoding and navigating linguistic structure.* PhD thesis, University of Maryland, College Park, MD.

Patil, U., Vasishth, S., and Lewis, R. L. (2016). Retrieval interference in syntactic processing: The case of reflexive binding in English. *Frontiers in Psychology*, 7(329):1–18.

Phillips, C. (2003). Linear order and constituency. *Linguistic inquiry*, 34(1):37–90.

Pickering, M. and Barry, G. (1991). Sentence processing without empty categories. *Language and Cognitive Processes*, 6(3):229–259.

Pickering, M. and Garrod, S. (2013). An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*, 36(4):329–347.

Price, C. J. (2012). A review and synthesis of the first 20 years of PET and fMRI studies of heard speech, spoken language and reading. *NeuroImage*, 62:816–847.

Prince, A. and Smolensky, P. (2002). Optimality theory: Constraint integration in generative grammar. Retrieved from doi:10.7282/T34M92MV on 2019-05-30, original Tech Report from 1993.

Pyke, A., West, R. L., and LeFevre, J.-A. (2007). On-line reference assignment for anaphoric and non-anaphoric nouns: A unified, memory-based model in ACT-R. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29, pages 1403–1408.

Reinhart, T. (1976). *The syntactic domain of anaphora.* PhD thesis, MIT, Cambridge, MA.

Reitter, D., Keller, F., and Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive Science*, 35:587–637.

Rosenbloom, P. S., Demski, A., and Ustun, V. (2016). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, 7(1):1–103.

Ryu, B.-R. (2014). Semantic constraints on multiple case marking in Korean. In Gerland, D., Horn, C., Latrouite, A., and Ortmann, A., editors, *Meaning and Grammar of Nouns and Verbs*, pages 77–112. Düsseldorf: University Press.

Sadler, L. and Nordlinger, R. (2010). Nominal juxtaposition in Australian languages: An LFG analysis. *Journal of Linguistics*, 46:415–452.

Salvucci, D. D. and Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, 115(1):101–130.

Schafer, A. J., Speer, S. R., Warren, P., and White, S. D. (2000). Intonational disambiguation in sentence production and comprehension. *Journal of Psycholinguistic Research*, 29(2):169–182.

Selkirk, E. O. (1984). *Phonology and syntax: the relation between sound and structure.* MIT Press.

Slattery, T. J., Sturt, P., Christianson, K., and Yoshida, M. (2013). Lingering misinterpretations of garden path sentences arise from competing syntactic representations. *Journal of Memory and Language*, 69:104–120.

Snedeker, J. and Trueswell, J. (2003). Using prosody to avoid ambiguity: Effects of speaker awareness and referential context. *Journal of Memory and Language*, 48:103–130.

Sohn, H.-M. (1994). *Korean.* London : Routledge, London.

Stabler, E. P. (2010). Computational perspectives on minimalism. In Boeckx, C., editor, *Oxford handbook of linguistic minimalism*, pages 617–641. Oxford University Press.

Stabler, E. P. (2013). Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science*, 5:611–633.

Steedman, M. (2000). *The syntactic process.* MIT Press, Cambridge, MA.

Stewart, T., Choo, F.-X., and Eliasmith, C. (2014). Sentence processing in spiking neurons: A biologically plausible left-corner parser. In *Proceedings of the Cognitive Science Society*, volume 36.

Stowe, L. A. (1986). Parsing WH-constructions: Evidence for on-line gap location. *Language and Cognitive Processes*, 1(3):227–245.

Stowe, L. A. (1992). The processing implementation of syntactic constraints: The sentence matching debate. In *Island Constraints: Theory, Acquisition and Processing*, pages 419–443. Kluwer Academic Publishers, Dordrecht.

Sturt, P. (2003). The time-course of the application of binding constraints in reference resolution. *Journal of Memory and Language*, 48:542–562.

Sturt, P. and Crocker, M. W. (1998). Generalized monotonicity for reanalysis models. In *Reanalysis in sentence processing*, chapter 11, pages 365–400. Kluwer, Dordrecht.

Toivonen, I. (2003). *Non-projecting words: A case study of Swedish particles.* Kluwer Academic Publishers, Dordrecht.

Traxler, M. J., Morris, R. K., and Seely, R. E. (2002). Processing subject and object relative clauses: evidence from eye movements. *Journal of Memory and Language*, 47(69-90).

Traxler, M. J. and Pickering, M. J. (1996). Plausibility and the processing of unbounded dependencies: An eye-tracking study. *Journal of Memory and Language*, 35:454–475.

Traxler, M. J., Pickering, M. J., and Clifton, Jr., C. (1998). Adjunct attachment is not a form of lexical ambiguity resolution. *Journal of Memory and Language*, 39:558–592.

Van Dyke, J. A. and Lewis, R. L. (2003). Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. *Journal of Memory and Language*, 49:285–316.

van Gompel, R. P. G. and Liversedge, S. P. (2003). The influence of morphological information on cataphoric pronoun assignment. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 29(1):128–139.

van Herten, M., Chwilla, D. J., and Kolk, H. H. (2006). When heuristics clash with parsing routines: ERP evidence for conflict monitoring in sentence perception. *Journal of Cognitive Neuroscience*, 18(7):1181–1197.

van Maanen, L., van Rijn, H., and Borst, J. P. (2009). Stroop and picture-word interference are two sides of the same coin. *Psychonomic Bulletin & Review*, 16(6):987–999.

van Rij, J. (2012). *Pronoun processing: Computational, behavioral, and psychophysiological studies in children and adults*. Number 108 in Groningen Dissertations in Linguistics. Rijksuniversiteit Groningen.

van Rij, J., van Rijn, H., and Hendriks, P. (2011). WM load influences the interpretation of referring expressions. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 67–75, Portland, Oregon. Association for Computational Linguistics.

van Rij, J., van Rijn, H., and Hendriks, P. (2012). How WM load influences pronoun interpretation. In *Proceedings of the 11th International Conference on Cognitive Modeling*, pages 101–102.

van Rijn, H. and Anderson, J. R. (2003). Modeling lexical decision as ordinary retrieval. In Detje, F., Doerner, D., and Schaub, H., editors, *Proceedings of the Fifth International Conference on Cognitive Modeling*, pages 207–212, Bamberg, Germany. Universitäts-Verlag Bamberg.

VanWagenen, S., Brennan, J., and Stabler, E. P. (2014). Quantifying parsing complexity as a function of grammar. In Schütze, C. T. and Stockall, L., editors, *Connectedness: Papers by and for Sarah VanWagenen*, volume 18 of *UCLA Working Papers in Linguistics*. Los Angeles: UCLA.

Wagers, M. W. (2013). Memory mechanisms for *wh*-dependency formation and their implications for islandhood. In Sprouse, J. and Hornstein, N., editors, *Experimental syntax and island effects*, chapter 7, pages 161–185. Cambridge University Press.

Wagers, M. W. and Phillips, C. (2014). Going the distance: Memory and control processes in active dependency construction. *The Quarterly Journal of Experimental Psychology*, 67(7):1274–1301.

Wagner, M. and Watson, D. G. (2010). Experimental and theoretical advances in prosody: A review. *Language and Cognitive Processes*, 25(7-9):905–945.

Watson, D. and Gibson, E. (2005). Intonational phrasing and constituency in language production and comprehension. *Studia Linguistica*, 59(2/3):279–300.

Wescoat, M. T. (1987). Practical instructions for working with the formalism of Lexical Functional Grammar. *Lexical-Functional Grammar. Course Material for L*, 1229:1987.

Wescoat, M. T. (2002). *On lexical sharing*. PhD thesis, Stanford University, Stanford, CA.

Yeon, J. and Brown, L. (2011). *Korean: A Comprehensive Grammar*. Abingdon, UK: Routledge.