Chapter 34

HPSG and Categorial Grammar

Yusuke Kubota
University of Tsukuba

This chapter aims to offer an up-to-date comparison of HPSG and categorial grammar (CG). Since the CG research itself consists of two major types of approaches with overlapping but distinct goals and research strategies, I start by giving an overview of these two variants of CG (section 2). This is followed by a comparison of HPSG and CG at a broad level, in terms of the general architecture of the theory (section 3), and then, by a somewhat more detailed comparison of specific linguistic analyses of some selected phenomena (section 4). The chapter ends by briefly touching on issues related to computational implementation and human sentence processing (section 5). Thoughtout the discussion, I attempt to highlight both the similarities and differences between HPSG and CG research, in the hope of stimulating further research in the two research communities on their respective open questions, and so that the two communities can continue to learn from each other.

1 Introduction

The goal of this chapter is to provide a comparison between HPSG and categorial grammar (CG). The two theories share certain important insights, mostly due to the fact that they are among the so-called 'lexicalist', 'non-transformational' theories of syntax that were proposed as major alternatives to the mainstream transformational syntax in the 1980s (see Borsley & Börjars (2011) for an overview of these theories). However, due to the differences in the main research goals in the respective communities in which these approaches have been developed, there are certain nontrivial differences between them as well. The present chapter assumes researchers working in HPSG or other non-CG theories of syntax as its main audience, and aims to inform them of key aspects of CG which

Yusuke Kubota. 2019. HPSG and Categorial Grammar. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 793–839. Berlin: Language Science Press. DOI:??

make it distinct from other theories of syntax. While computational implementation and investigations of the formal properties of grammatical theory have been important in both HPSG and CG research, I will primarily focus on the linguistic aspects in the ensuing discussion, with pointers to literature on mathematical and computational studies. Throughout the discussion, I presuppose basic familiarity with HPSG (with pointers to relevant chapters in the handbook). The present handbook contains chapters that compare HPSG with other grammatical theories, including the present one. I encourage the reader to take a look at the other theory comparison chapters too (as well as other chapters dealing with specific aspects of HPSG in greater detail), in order to obtain a fuller picture of the theoretical landscape in current (non-transformational) generative syntax research.

2 Two varieties of CG

CG is actually not a monolithic theory, but is a family of related approaches (or, perhaps more accurately, it is much *less of* a monolithic theory than either HPSG or LFG is). For this reason, I will start my discussion by sketching some important features of two major varieties of CG, Combinatory Categorial Grammar (CCG) (Steedman 2000; Steedman 2012) and Type-Logical Categorial Grammar (TLCG; or 'Type-Logical Grammar') (Morrill 1994; Moortgat 2011).[1] After presenting the 'core' component of CG that is shared between the two approaches—which is commonly referred to as the 'AB grammar'—I introduce aspects of the respective approaches in which they diverge from each other.

2.1 Notation and presentation

Before getting started, some comments are in order as to the notation and the mode of presentation adopted. Two choices are made for the notation. First, CCG and TLCG traditionally adopt different notations of the slash. I stick to the TLCG notation throughout this chapter for notational consistency. Second, I present all the fragments below in the so-called labelled deduction notation of (Prawitz-style) natural deduction. In particular, I follow Oehrle (1994) and Morrill (1994) in the use of 'term labels' in labelled deduction to represent prosodic and semantic information of linguistic expressions. This involves writing linguistic

[1] For more detailed introductions to these different variants of CG, see Steedman & Baldridge (2011) (on CCG) and Oehrle (2011) (on TLCG), both included in Borsley & Börjars (2011).

expressions as *tripartite signs*, formally, tuples of prosodic form, semantic interpretation and syntactic category (or syntactic type). Researchers familiar with HPSG should find this notation easy to read and intuitive; the idea is essentially the same as how linguistic signs are conceived of in HPSG. In the CG literature, this notation has its roots in the conception of 'multidimensional' linguistic signs in earlier work by Dick Oehrle (Oehrle 1988). But the reader should be aware that this is *not* the standard notation in which either CCG or TLCG is typically presented. Also, logically savvy readers may find this notation somewhat confusing since it (unfortunately) obscures certain aspects of CG pertaining to its logical properties. In any event, it is important to keep in mind that different notations co-exist in the CG literature (and the logic literature behind it) partly because of standard convention and practice in different (sub)fields and partly because of the need to highlight different aspects of the same formal system in different contexts. For the mode of presentation, the emphasis is consistently on linguistic (rather than computational or logical) aspects, with pointers to the relevant literature for readers interested in these other aspects. The presentation below moreover will not necessarily aim for historical accuracy, and I have chosen to gloss over certain minor differences for the sake of facilitating comparison among different subspecies of CG, and, more generally, between CG and HPSG (and other grammatical theories). The reader is therefore encouraged to consult primary sources for more accurate (and authoritative) presentations of each of the different subspecies of CG discussed below.

2.2 The AB grammar

I start with a simple fragment of CG called the AB GRAMMAR, consisting of just two syntactic rules in (1):

(1) a. FORWARD SLASH ELIMINATION

$$\frac{\mathsf{a};\ A/B \qquad \mathsf{b};\ B}{\mathsf{a} \circ \mathsf{b};\ A}\ /\mathrm{E}$$

b. BACKWARD SLASH ELIMINATION

$$\frac{\mathsf{b};\ B \qquad \mathsf{a};\ B\backslash A}{\mathsf{b} \circ \mathsf{a};\ A}\ \backslash\mathrm{E}$$

With a somewhat minimal lexicon in (7), we can license the sentence *John saw Mary* as in (3). The two slashes / and \ are used to form 'complex' syntactic categories (more on this below) indicating valence information: the transitive verb *loves* is assigned the category (NP\S)/NP since it first combines an NP to its right (i.e. the direct object) and then another NP to its right (i.e. the subject).

(2) a. john; NP

b. mary; NP

c. ran; NP\S

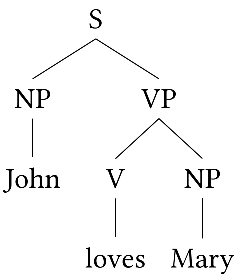
d. loves; (NP\S)/NP

(3)

$$\frac{\text{john; NP} \qquad \dfrac{\text{mary; NP} \quad \text{loves; }(\text{NP}\backslash\text{S})/\text{NP}}{\text{loves} \circ \text{mary; NP}\backslash\text{S}}{/\text{E}}}{\text{john} \circ \text{loves} \circ \text{mary; S}}{\backslash\text{E}}$$

At this point, this is just like the familiar PSG analysis of the following form, except that the symbol VP is replaced by NP\S:

(4)



Things will start looking more interesting as we make the fragment more complex (and also by adding the semantics), but before doing so I first introduce some basic assumptions, first on syntactic categories (below) and then on semantics (next section).

Syntactic categories (or syntactic types) are defined recursively in CG. This can be concisely written using the so-called 'BNC notation' as follows:[2, 3]

(5) a. BaseType := { N, NP, PP, S }

b. Type := BaseType | Type\ Type | Type/Type

In words, anything that is a BaseType is a Type, and any complex expression of form A\ B or A/B where A and B are both Types is a Type. To give some examples, the following expressions are syntactic types according to the definition in (5b):[4]

[2] See section 3.3 below for the treatment of syntactic features (such as those used for agreement). I ignore this aspect for the fragment developed below for the sake of exposition. The treatment of syntactic features (or its analog) is a relatively underdeveloped aspect of CG syntax literature, as compared to HPSG research (where the whole linguistic theory is built on the basis of a theory/formalism of complex feature structures). See section 3.3 below.

[3] Recognizing PP as a basic type is somewhat non-standard, although there does not seem to be any consensus on what should be regarded as a (reasonably complete) set of basic syntactic types for natural language syntax.

[4] I omit parentheses for a sequence of the same type of slash, for which disambiguation is obvious—for example, A\A\A is an abbreviation for (A\(A\A)).

(6)
 a. S\S
 b. (NP\S)/NP/NP
 c. (S/(NP\S))\(S/NP)
 d. ((NP\S)\(NP\S))\((NP\S)\(NP\S))

One important feature of CG is that, like HPSG, it lexicalizes the valence (or subcategorization) properties of linguistic expressions. Unlike HPSG, where this is done by a list (or set) valued syntactic feature, in CG, complex syntactic categories directly represent the combinatoric (i.e. valence) properties of lexical items. For example, lexical entries for intransitive and transitive verbs in English will look like the following (semantics is omitted here but will be supplied later):

(7)
 a. ran; NP\S
 b. read; (NP\S)/NP
 c. introduces; (NP\S)/PP/NP

(7a) says that the verb *ran* combines with its argument NP *to its left* to become an S. Likewise, (7c) says that *read* first combines with an NP *to its right* and then another NP to its left to become an S.

One point to keep in mind (though it may not seem to make much difference at this point) is that in CG, syntactic rules are thought of as logical rules and the derivations of sentences like (3) as *proofs* of the well-formedness of particular strings as sentences. From this 'logical' point of view, the two slashes should really be thought of as directional variants of implication (that is, both A/B and $B\backslash A$ essentially mean '*if* there is a B, *then* there is an A'), and the two rules of Slash Elimination introduced in (1) should be thought of as directional variants of MODUS PONENS ($B \rightarrow A, B \vdash A$). This analogy between natural language syntax and logic is emphasized in particular in the TLCG research.

2.3 Syntax-semantics interface in CG

One attractive property of CG as a theory of natural language syntax is its straightforward syntax-semantics interface. In particular, there is a functional mapping from syntactic categories to semantic types.[5] For the sake of exposition, I assume

[5]Technically, this is ensured in TLCG by the homomorphism from the syntactic type logic to the semantic type logic (the latter of which is often implicit) and the so-called Curry-Howard correspondence between proofs and terms.

ref to van Benthem

an extensional fragment of Montagovian model-theoretic semantics in what follows, but it should be noted that the CG syntax is neutral to the choice of the specific variant of semantics to go with it.[6]

Assuming the standard recursive definition of semantic types as in (8) (with basic types e (individuals) and t (truth values)), we can define the function Sem that returns, for each syntactic category given as input, its semantic type, as in (9) and (10).

(8) a. BaseSemType := { e, t }
b. SemType := BaseSemType | SemType $\rightarrow$ SemType

(9) (Base Case)
a. Sem(NP) = Sem(PP) = e
b. Sem(N) = $e \rightarrow t$
c. Sem(S) = t

(10) (Recursive Clause)
For any complex syntactic category of the form A/B (or $B\backslash A$),
$\mathsf{Sem}(A/B)$ (= $\mathsf{Sem}(B\backslash A)$) = $\mathsf{Sem}(B) \rightarrow \mathsf{Sem}(A)$

For example, we have $\mathsf{Sem}(\mathrm{S}/(\mathrm{NP}\backslash\mathrm{S})) = (e \rightarrow t) \rightarrow t$ (for subject position quantifier in CCG).

Syntactic rules with semantics can then be written as in (11) (where the semantic effect of these rules is FUNCTION APPLICATION) and a sample derivation with semantic annotation is given in (12).

(11) a. FORWARD SLASH ELIMINATION

$$\frac{\mathsf{a}; \mathcal{F}; A/B \quad \mathsf{b}; \mathcal{G}; B}{\mathsf{a} \circ \mathsf{b}; \mathcal{F}(\mathcal{G}); A}/\mathrm{E}$$

b. BACKWARD SLASH ELIMINATION

$$\frac{\mathsf{b}; \mathcal{G}; B \quad \mathsf{a}; \mathcal{F}; B\backslash A}{\mathsf{b} \circ \mathsf{a}; \mathcal{F}(\mathcal{G}); A}\backslash\mathrm{E}$$

(12)

$$\dfrac{\mathsf{john}; \mathbf{j}; \mathrm{NP} \quad \dfrac{\mathsf{loves}; \mathbf{love}; (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP} \quad \mathsf{mary}; \mathbf{m}; \mathrm{NP}}{\mathsf{loves} \circ \mathsf{mary}; \mathbf{love}(\mathbf{m}); \mathrm{NP}\backslash\mathrm{S}}/\mathrm{E}}{\mathsf{john} \circ \mathsf{loves} \circ \mathsf{mary}; \mathbf{love}(\mathbf{m})(\mathbf{j}); \mathrm{S}}\backslash\mathrm{E}$$

A system of CG with only the Slash Elimination rules like the fragment above is called the AB GRAMMAR, so called because it corresponds to the earliest form of CG formulated by Ajdukiewicz (1935) and Bar-Hillel (1953).

[6]See for example Martin (2013) and Bekki & Mineshima (2017) for recent proposals on adopting compositional variants of (hyper)intensional dynamic semantics and proof theoretic semantics, respectively, for the semantic component of CG-based theories of natural language.

2.4 Combinatory Categorial Grammar

2.4.1 An 'ABC' fragment: AB grammar with order-preserving 'combinatory' rules

To do some interesting linguistic analysis, we need some more machinery. I now extend the AB fragment above by adding two types of rules: TYPE-RAISING and (Harmonic) FUNCTION COMPOSITION. These are a subset of rules typically entertained in CCG. I call the resultant system ABC GRAMMAR (AB + Function Composition).[7] Though it is an impoverished version of CCG, it already enables an interesting and elegant analysis of NONCONSTITUENT COORDINATION (NCC), originally due to Steedman (1985) and Dowty (1988), which is essentially identical to the analysis of NCC in the current versions of both CCG and TLCG. I will then discuss the rest of the rules constituting CCG in the next section. The reason for drawing a distinction between the 'ABC' fragment and (proper) CCG is just for the sake of exposition. The rules introduced in the present section have the property that they are all derivable as *theorems* in the (associative) Lambek calculus, the calculus that underlies most variants of TLCG. For this reason, separating the two sets of rules helps clarify the similarities and differences between CCG and TLCG.

The TYPE RAISING and FUNCTION COMPOSITION rules are defined as in (13) and (14), respectively.

(13) a. Forward Function Composition

$$\frac{\mathsf{a};\ \mathcal{F};\ A/B \qquad \mathsf{b};\ \mathcal{G};\ B/C}{\mathsf{a} \circ \mathsf{b};\ \lambda x.\mathcal{F}(\mathcal{G}(x));\ A/C}\ \mathrm{FC}$$

b. Backward Function Composition

$$\frac{\mathsf{b};\ \mathcal{G};\ C\backslash B \qquad \mathsf{a};\ \mathcal{F};\ B\backslash A}{\mathsf{b} \circ \mathsf{a};\ \lambda x.\mathcal{F}(\mathcal{G}(x));\ C\backslash A}\ \mathrm{FC}$$

(14) a. Forward Type Raising

$$\frac{\mathsf{a};\ \mathcal{F};\ A}{\mathsf{a};\ \lambda v.v(\mathcal{F});\ B/(A\backslash B)}\ \mathrm{TR}$$

b. Backward Type Raising

$$\frac{\mathsf{a};\ \mathcal{F};\ A}{\mathsf{a};\ \lambda v.v(\mathcal{F});\ (B/A)\backslash B}\ \mathrm{TR}$$

The Type-Raising rules are essentially rules of 'type lifting' familiar in the formal semantics literature, except that they specify the 'syntactic effect' of type lifting explicitly. Similarly Function Composition rules can be understand as function composition in the usual sense (as in mathematics and functional programming), except, again, that the syntactic effects are explicitly specified.

[7] This is not a standard terminology, giving a name to this fragment is convenient for the purpose of the discussion below.

As noted by Steedman (1985), with Type Raising and Function Composition, we can analyze a string of words such as *John loves* as a constituent of type S/NP, that is, an expression that is looking for an NP to its right to become an S:

(15)
$$\dfrac{\dfrac{\text{john}; \mathbf{j}; \mathrm{NP}}{\text{john}; \lambda f.f(\mathbf{j}); \mathrm{S/(NP\backslash S)}}\,\mathrm{TR} \qquad \text{loves}; \mathbf{love}; \mathrm{(NP\backslash S)/NP}}{\text{john} \circ \text{loves}; \lambda x.\mathbf{love}(x)(\mathbf{j}); \mathrm{S/NP}}\,\mathrm{FC}$$

Assuming generalized conjunction (with the standard definition for the generalized conjunction operator $\sqcap$ *a la* Partee & Rooth (1983) and the polymorphic syntactic category $(X\backslash X)/X$ for *and*, the analysis for a right-node raising (RNR) sentence such as (16) is straightforward, as in (17).

(16) John loves, and Bill hates, Mary.

(17)
$$\dfrac{\dfrac{\begin{matrix}\vdots \\ \text{john} \circ \text{loves}; \\ \lambda x.\mathbf{love}(x)(\mathbf{j}); \mathrm{S/NP}\end{matrix} \qquad \dfrac{\begin{matrix}\text{and}; \\ \sqcap; (X\backslash X)/X\end{matrix} \qquad \begin{matrix}\vdots \\ \text{bill} \circ \text{hates}; \\ \lambda x.\mathbf{hate}(x)(\mathbf{b}); \mathrm{S/NP}\end{matrix}}{\begin{matrix}\text{and} \circ \text{bill} \circ \text{hates}; \\ \sqcap(\lambda x.\mathbf{hate}(x)(\mathbf{b})); \mathrm{(S/NP)\backslash(S/NP)}\end{matrix}}\,\mathrm{FA}}{\text{john} \circ \text{loves} \circ \text{and} \circ \text{bill} \circ \text{hates}; (\lambda x.\mathbf{love}(x)(\mathbf{j})) \sqcap (\lambda x.\mathbf{hate}(x)(\mathbf{b})); \mathrm{S/NP}}\,\mathrm{FA} \qquad \begin{matrix}\text{mary}; \\ \mathbf{m}; \mathrm{NP}\end{matrix}}{\text{john} \circ \text{loves} \circ \text{and} \circ \text{bill} \circ \text{hates} \circ \text{mary}; \mathbf{love}(\mathbf{m})(\mathbf{j}) \wedge \mathbf{hate}(\mathbf{m})(\mathbf{b}); \mathrm{S}}\,\mathrm{FA}$$

Dowty (1988) showed that this analysis extends straightforwardly to the (slightly) more complex case of argument cluster coordination (ACC), such as (18), as in (19) (here, VP, TV and DTV are abbreviations of NP\S, (NP\S)/NP, and (NP\S)/NP/NP, respectively).

(18) Mary gave Bill the book and John the record.

(19)
$$\dfrac{\begin{matrix}\text{mary}; \\ \mathbf{m}; \\ \mathrm{NP}\end{matrix} \qquad \dfrac{\begin{matrix}\text{gave}; \\ \mathbf{give}; \\ \mathrm{DTV}\end{matrix} \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}\text{bill}; \\ \mathbf{b}; \mathrm{NP}\end{matrix}}{\begin{matrix}\text{bill}; \\ \lambda P.P(\mathbf{b}); \\ \mathrm{DTV\backslash TV}\end{matrix}}\,\mathrm{TR} \qquad \dfrac{\begin{matrix}\text{the} \circ \text{book}; \\ \iota(\mathbf{bk}); \mathrm{NP}\end{matrix}}{\begin{matrix}\text{the} \circ \text{book}; \\ \lambda Q.Q(\iota(\mathbf{bk})); \\ \mathrm{TV\backslash VP}\end{matrix}}\,\mathrm{TR}}{\begin{matrix}\text{bill} \circ \text{the} \circ \text{book}; \\ \lambda R.R(\mathbf{b})(\iota(\mathbf{bk})); \mathrm{DTV\backslash VP}\end{matrix}}\,\mathrm{FC} \qquad \dfrac{\begin{matrix}\text{and}; \\ \sqcap; \\ (X\backslash X)/X\end{matrix} \qquad \begin{matrix}\vdots \\ \text{john} \circ \text{the} \circ \text{record}; \\ \lambda R.R(\mathbf{j})(\iota(\mathbf{rc})); \\ \mathrm{DTV\backslash VP}\end{matrix}}{\begin{matrix}\text{and} \circ \text{john} \circ \text{the} \circ \text{record}; \\ \sqcap(\lambda R.R(\mathbf{j})(\iota(\mathbf{rc}))); \\ \mathrm{(DTV\backslash VP)\backslash(DTV\backslash VP)}\end{matrix}}}{\begin{matrix}\text{bill} \circ \text{the} \circ \text{book} \circ \text{and} \circ \text{john} \circ \text{the} \circ \text{record}; \\ \lambda R.R(\mathbf{j})(\iota(\mathbf{rc})) \sqcap R(\mathbf{b})(\iota(\mathbf{bk})); \mathrm{DTV\backslash VP}\end{matrix}}}{\begin{matrix}\text{gave} \circ \text{bill} \circ \text{the} \circ \text{book} \circ \text{and} \circ \text{john} \circ \text{the} \circ \text{record}; \\ \mathbf{give}(\mathbf{j})(\iota(\mathbf{rc})) \sqcap \mathbf{give}(\mathbf{b})(\iota(\mathbf{bk})); \mathrm{VP}\end{matrix}}}{\begin{matrix}\text{mary} \circ \text{gave} \circ \text{bill} \circ \text{the} \circ \text{book} \circ \text{and} \circ \text{john} \circ \text{the} \circ \text{record}; \\ \mathbf{give}(\mathbf{j})(\iota(\mathbf{rc}))(\mathbf{m}) \wedge \mathbf{give}(\mathbf{b})(\iota(\mathbf{bk}))(\mathbf{m}); \mathrm{S}\end{matrix}}$$

Here, by Type Raising, the indirect and direct objects become functions that can be combined via Function Composition, to form a non-standard constituent that can be coordinated. After two such expressions are conjoined, the verb is fed as an argument to return a VP. Intuitively, the idea behind this analysis is that *Bill the book* is of type DTV\VP since if it were to combine with an actual ditransitive verb (such as *gave*), we would obtain a VP (*gave Bill a book*). Note that in both the RNR and ACC examples above, the right semantic interpretation for the whole sentence is assigned compositionally via the general definitions of type raising and function composition given above.

2.4.2 From ABC to CCG

CCG is a version of CG developed by Mark Steedman since the 1980s with extensive linguistic application. The best sources for CCG are the three books by Steedman (Steedman 1997; 2000; Steedman 2012), which present treatments of major linguistic phenomena in CCG, and gives pointers to earlier literature. CCG is essentially a rule-based extension of the AB grammar. We have already seen in the previous section the two key components that constitute this extension: Type Raising and (Harmonic) Function Composition.[8] There are aspects of natural language syntax that cannot be handled adequately in this simple system, and in such situations, CCG makes (restricted) use of additional rules. This point can be illustrated nicely with two issues that arise in connection with the analysis of long-distance dependencies.

The basic idea behind the CCG analysis of long-distance dependencies, due originally to Ades & Steedman (1982), is very simple and is similar in spirit to the HPSG analysis in terms of SLASH feature percolation. Specifically, CCG analyzes extraction dependency via a chain of function composition, as illustrated by the derivation for (20) in (21).

(20) This is the book that John thought that Mary read __.

[8] There is actually a subtle point about Type Raising rules. It seems that recent versions of CCG (Steedman 2012) do not take them to be syntactic rules but rather assume that Type Raising is an operation in the lexicon, due to parsing considerations. It is also worth noting in this connection that the CCG-based syntactic fragment that Jacobson (1999; 2000) assumes for her Variable-Free Semantics is actually a quite different system from Steedman's version of CCG in that it crucially assumes Geach rules (which increases the complexity of the syntactic type of the linguistic expression) in the syntactic component.

(21)

$$\dfrac{\text{mary};\ \mathbf{m};\ \mathrm{NP}}{\text{mary};\ \lambda P.P(\mathbf{m});\ \mathrm{S/(NP\backslash S)}}\ \mathrm{TR}$$

$$\dfrac{\text{mary};\ \lambda P.P(\mathbf{m});\ \mathrm{S/(NP\backslash S)} \qquad \text{read};\ \mathbf{read};\ \mathrm{(NP\backslash S)/NP}}{\text{mary} \circ \text{read};\ \lambda x.\mathbf{read}(x)(\mathbf{m});\ \mathrm{S/NP}}\ \mathrm{FC}$$

$$\dfrac{\text{that};\ \lambda p.p;\ \mathrm{S'/S} \qquad \text{mary} \circ \text{read};\ \lambda x.\mathbf{read}(x)(\mathbf{m});\ \mathrm{S/NP}}{\text{that} \circ \text{mary} \circ \text{read};\ \lambda x.\mathbf{read}(x)(\mathbf{m});\ \mathrm{S'/S}}\ \mathrm{FC}$$

$$\dfrac{\text{thought};\ \mathbf{think};\ \mathrm{(NP\backslash S)/S'} \qquad \text{that} \circ \text{mary} \circ \text{read};\ \lambda x.\mathbf{read}(x)(\mathbf{m});\ \mathrm{S'/S}}{\text{thought} \circ \text{that} \circ \text{mary} \circ \text{read};\ \lambda x.\mathbf{think}(\mathbf{read}(x)(\mathbf{m}));\ \mathrm{(NP\backslash S)/NP}}\ \mathrm{FC}$$

$$\dfrac{\text{john};\ \mathbf{j};\ \mathrm{NP}}{\text{john};\ \lambda P.P(\mathbf{j});\ \mathrm{S/(NP\backslash S)}}\ \mathrm{TR}$$

$$\dfrac{\text{john};\ \lambda P.P(\mathbf{j});\ \mathrm{S/(NP\backslash S)} \qquad \text{thought} \circ \text{that} \circ \text{mary} \circ \text{read};\ \lambda x.\mathbf{think}(\mathbf{read}(x)(\mathbf{m}));\ \mathrm{(NP\backslash S)/NP}}{\text{john} \circ \text{thought} \circ \text{that} \circ \text{mary} \circ \text{read};\ \lambda x.\mathbf{think}(\mathbf{read}(x)(\mathbf{m}))(\mathbf{j});\ \mathrm{S/NP}}\ \mathrm{FC}$$

$$\dfrac{\text{that};\ \lambda P\lambda Q\lambda x.\ Q(x) \wedge P(x);\ \mathrm{(N\backslash N)/(S/NP)} \qquad \text{john} \circ \text{thought} \circ \text{that} \circ \text{mary} \circ \text{read};\ \lambda x.\mathbf{think}(\mathbf{read}(x)(\mathbf{m}))(\mathbf{j});\ \mathrm{S/NP}}{\text{that} \circ \text{john} \circ \text{thought} \circ \text{that} \circ \text{mary} \circ \text{read};\ \lambda Q\lambda x.Q(x) \wedge \mathbf{think}(\mathbf{read}(x)(\mathbf{m}))(\mathbf{j});\ \mathrm{N\backslash N}}\ \mathrm{FA}$$

Like (many versions of) HPSG, CCG does not assume any empty expression at the gap site. Instead, the information that the subexpression such as *Mary read* and *thought that Mary read* are missing an NP on the right edge is encoded in the syntactic category of the linguistic expression. *Mary read* is assigned the type S/NP Since it is a sentence missing an NP on its right edge. *thought that Mary read* is of type VP/NP Since it is a VP missing an NP on its right edge, etc. Expressions that are not originally functions (such as the subject NPs in the higher and lower clauses inside the relative clause in (20)) are first type-raised. Then, function composition effectively 'delay' the saturation of the object NP argument of the embedded verb, until the whole relative clause meets the relative pronoun, which itself is a higher-order function that takes a sentence missing an NP (of type S/NP).

The successive passing of the /NP specification to larger structures is essentially analogous to the treatment of extraction via the SLASH feature in HPSG. However, unlike HPSG, which has a dedicated feature that handles this information passing, CCG achieves the effect via the ordinary slash that is also used for local syntactic composition.[9]

This difference immediately raises some issues for the CCG analysis of extraction. First, in (20), the NP gap happens to be on the right edge of the sentence, but this is not always the case. Harmonic function composition alone cannot handle non-peripheral extraction of the sort found in examples such as the following:

(22) This is the book that John thought that [Mary read __ at school].

[9] There are several complications that arise in the CCG analysis because of this architectural difference, which I discuss immediately below. But one possible welcome consequence of the CCG approach is that the quite complex set of specifications of SLASH feature percolation (including the *termination* of percolation) are all unnecessary in CCG.

Assuming that *at school* is a VP modifier of type (NP\S)\(NP\S), what is needed here is a mechanism that assigns the type VP/NP to the string *read* __ *at school*, despite the fact that the missing NP is not on the right edge. CCG employs a special rule of 'mixed' function composition for this purpose, defined as follows:

(23) Crossed Function Composition

$$\frac{\mathsf{a};\ \mathcal{G};\ A/B \qquad \mathsf{b};\ \mathcal{F};\ A\backslash C}{\mathsf{a} \circ \mathsf{b};\ \lambda x.\mathcal{F}(\mathcal{G}(x));\ C/B}\ \mathrm{xFC}$$

Unlike the harmonic counterpart of (23) (in which a has the type $B\backslash A$), the directionality of the premise is different in the two premises, and the resultant category inherits the slash of the main functor (i.e. $B\backslash C$).

Once this non-order-preserving version of function composition is introduced in the grammar, the derivation for (22) is straightforward, as in (24):

(24)

$$\dfrac{\dfrac{\mathsf{mary};\ \mathbf{m};\ \mathrm{NP}}{\mathsf{mary};\ \lambda P.P(\mathbf{m});\ \mathrm{S/(NP\backslash S)}}\mathrm{TR} \qquad \dfrac{\mathsf{read};\ \mathbf{read};\ \mathrm{(NP\backslash S)/NP} \qquad \mathsf{at} \circ \mathsf{school};\ \mathbf{at\text{-}school};\ \mathrm{(NP\backslash S)\backslash(NP\backslash S)}}{\mathsf{read} \circ \mathsf{at} \circ \mathsf{school};\ \lambda x.\mathbf{at\text{-}school}(\mathbf{read}(x));\ \mathrm{(NP\backslash S)/NP}}\mathrm{xFC}}{\mathsf{mary} \circ \mathsf{read} \circ \mathsf{at} \circ \mathsf{school};\ \lambda x.\mathbf{at\text{-}school}(\mathbf{read}(x))(\mathbf{m});\ \mathrm{S/NP}}\mathrm{FC}$$

Unless appropriately constrained, the addition of the crossed composition rule leads to overgeneration, since non-extracted expressions cannot change word order so freely in English. For example, without additional restrictions, the simple CCG fragment above overgenerates examples such as the following:

(25) *Mary read at school it.

Here, I will not go into the technical details of how this issue is addressed in the CCG literature.[10] A broad consensus in the contemporary CCG literature seems to be to regulate the syntactic contexts in which special rules such as crossed composition in (23) are licensed via the notion of 'structural control' inherited to CCG from the 'multi-modal' variant of TLCG (see Baldridge (2002) and Steedman & Baldridge (2011)).

Another issue that arises in connection to extraction is how to treat multiple gaps corresponding to a single filler. The simple fragment developed above cannot license examples involving parasitic gaps such as the following:[11]

[10] The word order in (25) is of course fine in examples instantiating Heavy NP Shift.

[11] Multiple gaps in coordination (i.e. ATB extraction) is not an issue, since these cases can be handled straightforwardly via the polymorphic definition of generalized conjunction in CCG, in just the same way that unsaturated shared arguments in each conjunct are identified with one another.

(26) a. This is the article that I filed __ without reading __.

b. Peter is a guy who even the best friends of __ think __ should be closely watched.

Since neither type raising nor function composition changes the number of 'gaps' passed on to a larger expression, we need a new mechanism here. **steedman1987** proposes the following rule to deal with this issue:

(27) Substitution

$$\frac{\mathsf{a};\ \mathcal{G};\ A/B \qquad \mathsf{b};\ \mathcal{F};\ (A\backslash C)/B}{\mathsf{a} \circ \mathsf{b};\ \lambda x.\mathcal{F}(x)(\mathcal{G}(x));\ C/B}\,\mathrm{S}$$

This 'substitution' rule has the effect of 'collapsing' the arguments of the two inputs into one, to be saturated by a single filler. The derivation for the adjunct parasitic gap example in (26a) then goes as follows (where VP is an abbreviation for NP\S):

(28)
$$\dfrac{\begin{array}{c}\mathsf{filed};\\ \mathbf{file};\ \mathrm{VP/NP}\end{array} \quad \dfrac{\begin{array}{c}\mathsf{without};\\ \mathbf{wo};\ (\mathrm{VP\backslash VP})/\mathrm{VP}\end{array} \quad \begin{array}{c}\mathsf{reading};\\ \mathbf{read};\ \mathrm{VP/NP}\end{array}}{\begin{array}{c}\mathsf{without} \circ \mathsf{reading};\\ \lambda x.\mathbf{wo}(\mathbf{read}(x));\ (\mathrm{VP\backslash VP})/\mathrm{NP}\end{array}}\mathrm{FC}}{\begin{array}{c}\mathsf{filed} \circ \mathsf{without} \circ \mathsf{reading};\\ \lambda x.\mathbf{wo}(\mathbf{read}(x))(\mathbf{file}(x));\ \mathrm{VP/NP}\end{array}}\mathrm{S}$$

Like the crossed composition rule, the availability of the substitution rule should be restricted to extraction environments. In earlier versions of CCG, this was taken care of via a stipulation on the rule itself. Baldridge (2002) proposed an improvement of the organization of the CCG rule system in which the applicability of particular rules is governed by lexically specified 'modality' encodings. See Steedman & Baldridge (2011) for this relatively recent development in CCG.

2.5 Type-Logical Categorial Grammar

The 'rule-based' nature of CCG should be clear from the above exposition. TLCG takes a somewhat different perspective on the underlying architecture of the grammar of natural language. Specifically, in TLCG, the rule system of grammar is literally taken to be a kind of logic. Consequently, all grammar rules are logical inference rules reflecting the properties of (typically a small number of) logical connectives such as / and \ (which are viewed as directional variants of implication). This conceptual shift can be best illustrated by first replacing the ABC grammar introduced in section 2.4.1 by the LAMBEK CALCULUS, where all the rules posited as primitive rules in the former are derived as *theorems* (in the technical sense of the term) in the latter.

Before moving on, I should hasten to note that the TLCG literature is more varied than the CCG literature, involving multiple related but distinct approaches. I choose to present one particular variant known as Hybrid TLCG in what follows, in line with the present chapter's linguistic emphasis. A brief comparison with major alternatives is offered in section 2.5.3. Other variants of TLCG, most notably, the Categorial Type Logics (Moortgat 2011) and Displacement Calculus (Morrill 2010) emphasize the logical and computational aspects. Moot & Retoré (2012) is a good introduction to TLCG covering the logical and computational foundations. For readers interested in linguistic application, Carpenter (1998) is a very valuable source for TLCG analyses of a wide range of linguistic phenomena at the syntax-semantics interface, with very lucid exposition.

2.5.1 The Lambek calculus

In addition to the Slash Elimination rules, which are identical to the two rules in the AB grammar from section 2.2, the Lambek calculus posits the Slash Introduction rules, which can be written in the current labelled deduction format as in (30).[12]

(29) a. Forward Slash Elimination

$$\frac{\mathsf{a};\ \mathcal{F};\ A/B \qquad \mathsf{b};\ \mathcal{G};\ B}{\mathsf{a} \circ \mathsf{b};\ \mathcal{F}(\mathcal{G});\ A}\ /\mathrm{E}$$

b. Backward Slash Elimination

$$\frac{\mathsf{b};\ \mathcal{G};\ B \qquad \mathsf{a};\ \mathcal{F};\ B\backslash A}{\mathsf{b} \circ \mathsf{a};\ \mathcal{F}(\mathcal{G});\ A}\ \backslash\mathrm{E}$$

(30) a. Forward Slash Introduction

$$\cfrac{\cfrac{\begin{array}{ccc} \vdots & \underline{[\varphi;\ x;\ A]^n} & \vdots \\ \vdots & \vdots & \vdots \end{array}}{\mathsf{b} \circ \varphi;\ \mathcal{F};\ B}}{\mathsf{b};\ \lambda x.\mathcal{F};\ B/A}\ /\mathrm{I}^n$$

b. Backward Slash Introduction

$$\cfrac{\cfrac{\begin{array}{ccc} \vdots & \underline{[\varphi;\ x;\ A]^n} & \vdots \\ \vdots & \vdots & \vdots \end{array}}{\varphi \circ \mathsf{b};\ \mathcal{F};\ B}}{\mathsf{b};\ \lambda x.\mathcal{F};\ A\backslash B}\ \backslash\mathrm{I}^n$$

I illustrate the workings of these rules in some examples first and then come back to some formal and notational issues. One consequence that immediately follows is that Type Raising and Function Composition (as well as other theorems; see, e.g., Jäger (2005: sec 2.2.5, pp. 46–)) are now derivable as theorems. As an illustration, the proofs for (14a) and (13a) are shown in (31) and (32), respectively.

[12] Morrill (1994) was the first to recast the Lambek calculus in this labelled deduction format.

(31) $$\dfrac{\dfrac{[\varphi; v; A\backslash B]^1 \qquad \mathsf{a}; \mathcal{F}; A}{\mathsf{a}\circ\varphi; v(\mathcal{F}); B}\,\backslash\mathrm{E}}{\mathsf{a}; \lambda v.v(\mathcal{F}); B/(A\backslash B)}\,/\mathrm{I}^1$$

(32) $$\dfrac{\dfrac{\mathsf{a}; \mathcal{F}; A/B \qquad \dfrac{[\varphi; x; C]^1 \qquad \mathsf{b}; \mathcal{G}; B/C}{\mathsf{b}\circ\varphi; \mathcal{G}(x); B}\,/\mathrm{E}}{\mathsf{a}\circ\mathsf{b}\circ\varphi; \mathcal{F}(\mathcal{G}(x)); A}\,/\mathrm{E}}{\mathsf{a}\circ\mathsf{b}; \lambda x.\mathcal{F}(\mathcal{G}(x)); A/C}\,/\mathrm{I}^1$$

These are just formal theorems, but they intuitively make sense. For example, what's going on in (32) is simple. We first hypothetically assume the existence of some expression of type C and combine it with B/C. This gives us a larger expression of type B, which then can be fed as an argument to A/B. At that point, we withdraw the initial hypothesis and conclude that what we really had was just something that would become an A *if* there is a C to its right, namely, an expression of type A/C. Thus, a sequence of expression of types A/B and B/C is proven to be of type A/C. This type of proof is known as HYPOTHETICAL REASONING, since it involves a step of positing a hypothesis initially and withdrawing that hypothesis at some later point.

Getting back to some notational issues, there are two crucial things to keep in mind about the notational convention adopted here (which I implicitly assumed above). First, the connective $\circ$ in the prosodic component designates string concatenation and is associative in both directions (i.e. $(\varphi_1 \circ \varphi_2) \circ \varphi_3 \equiv \varphi_1 \circ (\varphi_2 \circ \varphi_3)$). In other words, hierarchical structure is irrelevant for the prosodic representation. Thus, the prosodic variable φ in (32) appears at the right on the left periphery in the prosodic representation $\mathsf{a} \circ \mathsf{b} \circ \varphi$ on the condition of Forward Slash Introduction (30a). Note in particular that the application of the Introduction rules is conditioned on the position of the prosodic variable, and not on the position of the hypothesis itself in the proof tree (this latter convention is more standardly adopted when the Lambek calculus is presented in Prawitz-style natural deduction, though the two are equivalent—see, e.g., Carpenter (1998) and Jäger (2005)).

Hypothetical reasoning with Slash Introduction makes it possible to recast the CCG analysis of nonconstituent coordination we saw above within the logic of / and \. This reformulation fully retains the essential analytic ideas of the original CCG analysis but makes the underlying 'logic' of how bits and pieces are combined in this construction more transparent.

The following derivation illustrates how the 'reanalysis' of the string *Bill the book* as a derived constituent of the same type (VP/NP/NP)\VP as in (19) can be obtained in the Lambek calculus:

(33)
$$\dfrac{\dfrac{\dfrac{[\varphi; f; \mathrm{VP/NP/NP}]^1 \quad \textsf{bill}; \mathbf{b}; \mathrm{NP}}{\varphi \circ \textsf{bill}; f(\mathbf{b}); \mathrm{VP/NP}}{/\mathrm{E}} \quad \textsf{the} \circ \textsf{book}; \iota(\mathbf{bk}); \mathrm{NP}}{\varphi \circ \textsf{bill} \circ \textsf{the} \circ \textsf{book}; f(\mathbf{b})(\iota(\mathbf{bk})); \mathrm{VP}}{/\mathrm{E}}}{\textsf{bill} \circ \textsf{the} \circ \textsf{book}; \lambda f.f(\mathbf{b})(\iota(\mathbf{bk})); (\mathrm{VP/NP/NP})\backslash\mathrm{VP}}{\backslash\mathrm{I}^1}$$

At this point, one may wonder what the relationship is between the analysis of nonconstituent coordination via Type Raising and Function Composition in the ABC grammar in section 2.4.1 and the hypothetical reasoning-based analysis in the Lambek calculus just presented. Intuitively, they seem to achieve the same effect in slightly different ways. The logic-based perspective of TLCG allows us to obtain a deeper understanding of the relationship between them. To facilitate comparison, I first recast the TR+FC analysis in the Lambek calculus. The relevant part is the part that derives the 'noncanonical constituent' *Bill the book*:

(34)
$$\dfrac{\dfrac{\dfrac{[\varphi_3; R; \mathrm{DTV}]^3 \quad \dfrac{\dfrac{[\varphi_2; P; \mathrm{DTV}]^2 \quad \textsf{bill}; \mathbf{b}; \mathrm{NP}}{\varphi_2 \circ \textsf{bill}; P(\mathbf{b}); \mathrm{TV}}{/\mathrm{E}}}{\textsf{bill}; \lambda P.P(\mathbf{b}); \mathrm{DTV}\backslash\mathrm{TV}}{\backslash\mathrm{I}^2}}{\varphi_3 \circ \textsf{bill}; R(\mathbf{b}); \mathrm{TV}}{\backslash\mathrm{E}} \quad \dfrac{\dfrac{[\varphi_1; Q; \mathrm{TV}]^1 \quad \textsf{the} \circ \textsf{book}; \iota(\mathbf{bk}); \mathrm{NP}}{\varphi_1 \circ \textsf{the} \circ \textsf{book}; Q(\iota(\mathbf{bk})); \mathrm{VP}}{/\mathrm{E}}}{\textsf{the} \circ \textsf{book}; \lambda Q.Q(\iota(\mathbf{bk})); \mathrm{TV}\backslash\mathrm{VP}}{\backslash\mathrm{I}^1}}{\varphi_3 \circ \textsf{bill} \circ \textsf{the} \circ \textsf{book}; R(\mathbf{b})(\iota(\mathbf{bk})); \mathrm{VP}}{\backslash\mathrm{E}}}{\textsf{bill} \circ \textsf{the} \circ \textsf{book}; \lambda R.R(\mathbf{b})(\iota(\mathbf{bk})); \mathrm{DTV}\backslash\mathrm{VP}}{\backslash\mathrm{I}^3}$$

By comparing (34) and (33), we see that (34) contains some redundant steps. First, hypothesis 2 is introduced only to be replaced by hypothesis 3. This is completely redundant since we could have obtained exactly the same result by directly combining hypothesis 3 with the NP *Bill*. Similarly, hypothesis 1 can be eliminated by replacing it with the TV $\varphi_3 \circ$ book on the left-hand side of the third line from the bottom. By making these two simplifications, we obtain the derivation in (33).

The relationship between the more complex proof in (34) and the simpler one in (33) is parallel to the relationship between an unreduced lambda term (such as $\lambda P[P(\mathbf{b})](R)]$) and its β-normal form (i.e. $R(\mathbf{b})$). In fact, there is a formally precise one-to-one relationship between linear logic (of which the Lambek calculus is known to be a straightforward extension) and the typed lambda calculus known as the Curry-Howard Isomorphism ***[ref]***, according to which the lambda term that represents the proof (34) β-reduces to the term that represents the proof (33).[13] Technically, this is known as PROOF NORMALIZATION (Jäger (2005) contains

[13]There is a close relationship between these lambda terms representing proofs and the lambda terms that we write to notate semantics translations, especially if we write the latter at each step of derivation *without* performing β reduction. But it should be noted that these are distinct things.

a particularly useful discussion on this notion).

Thus, the logic-based architecture of the Lambek calculus (and various versions of TLCG, which are all extensions of the Lambek calculus) enables us to say, in a technically precise way, how (34) and (33) are the 'same' (or, more precisely, equivalent), by building on independently established results in mathematical logic and computer science. This is one big advantage of taking seriously the view, advocated by the TLCG research, that 'language *is* logic'.

2.5.2 Extending the Lambek calculus

Hypothetical reasoning is a very powerful (yet systematic) tool, but with forward and backward slashes, it is only good for analyzing expressions missing some material at the (right or left) periphery. This is problematic in the analyses of many linguistic phenomena, such as *wh*-extraction (where the 'gap' can be in a sentence-medial position) and quantifier scope (where the quantifier needs to 'covertly' move from a sentence-medial position), as well as various kinds of discontinuous constituency phenomena (see for example Morrill et al. (2011), which contains analyses of various types of discontinuous constituency phenomena in a recent version of TLCG known as 'Displacement Calculus'). In what follows, I sketch one particular, relatively recent approach to this problem, known as Hybrid Type-Logical Categorial Grammar (Hybrid TLCG; Kubota 2010; 2015; Kubota & Levine 2015). This approach combines the Lambek calculus with Oehrle's (1994) term-labelled calculus, which deals with discontinuity via by employing λ-binding in the prosodic component.

Hybrid TLCG extends the Lambek calculus with the Elimination and Introduction rules for the vertical slash:

(35) a. Vertical Slash Introduction

$$\dfrac{\dfrac{\begin{array}{ccc} \vdots & \underline{[\varphi; x; A]^n} & \vdots \\ \vdots & \vdots & \vdots \end{array}}{\mathsf{b}; \mathcal{F}; B}}{\lambda\varphi.\mathsf{b}; \lambda x.\mathcal{F}; B\upharpoonright A}\upharpoonright\mathrm{I}^n$$

b. Vertical Slash Elimination

$$\dfrac{\mathsf{a}; \mathcal{F}; A\upharpoonright B \quad \mathsf{b}; \mathcal{G}; B}{\mathsf{a}(\mathsf{b}); \mathcal{F}(\mathcal{G}); A}\upharpoonright\mathrm{E}$$

These rules allow us to model what (roughly) corresponds to covert movement in derivational frameworks. This is illustrated in (36) for the $\forall > \exists$ reading for the sentence *Someone talked to everyone today*:

(36)

$$\dfrac{\begin{array}{l}\textsf{talked} \circ \textsf{to};\\ \textbf{talked-to};\\ (\mathrm{NP}\backslash \mathrm{S})/\mathrm{NP}\end{array} \qquad \left[\begin{array}{l}\varphi_1;\\ x;\\ \mathrm{NP}\end{array}\right]^1}{\begin{array}{l}\textsf{talked} \circ \textsf{to} \circ \varphi_1;\\ \textbf{talked-to}(x); \mathrm{NP}\backslash \mathrm{S}\end{array}}/\mathrm{E}$$

$$\dfrac{\left[\begin{array}{l}\varphi_2;\\ y;\\ \mathrm{NP}\end{array}\right]^2 \qquad \begin{array}{l}\textsf{talked} \circ \textsf{to} \circ \varphi_1;\\ \textbf{talked-to}(x); \mathrm{NP}\backslash \mathrm{S}\end{array}}{\begin{array}{l}\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1;\\ \textbf{talked-to}(x)(y); \mathrm{S}\end{array}}\backslash \mathrm{E}$$

$$\dfrac{\begin{array}{l}\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1;\\ \textbf{talked-to}(x)(y); \mathrm{S}\end{array} \qquad \begin{array}{l}\textsf{today};\\ \textbf{tdy};\\ \mathrm{S}\backslash \mathrm{S}\end{array}}{\begin{array}{l}\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \textbf{tdy}(\textbf{talked-to}(x)(y)); \mathrm{S}\end{array}}\backslash \mathrm{E}$$

$$\dfrac{\begin{array}{l}\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \textbf{tdy}(\textbf{talked-to}(x)(y)); \mathrm{S}\end{array}}{\begin{array}{l}\lambda\varphi_2.\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \lambda y.\textbf{tdy}(\textbf{talked-to}(x)(y)); \mathrm{S}\upharpoonright\mathrm{NP}\end{array}}\upharpoonright\mathrm{I}^2$$

$$\dfrac{\begin{array}{l}\lambda\sigma.\sigma(\textsf{someone});\\ \text{Ǝ}_{\text{person}};\\ \mathrm{S}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\end{array} \qquad \begin{array}{l}\lambda\varphi_2.\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \lambda y.\textbf{tdy}(\textbf{talked-to}(x)(y)); \mathrm{S}\upharpoonright\mathrm{NP}\end{array}}{\begin{array}{l}\textsf{someone} \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \text{Ǝ}_{\text{person}}(\lambda y.\textbf{tdy}(\textbf{talked-to}(x)(y))); \mathrm{S}\end{array}}\upharpoonright\mathrm{E}$$

$$\dfrac{\begin{array}{l}\textsf{someone} \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \text{Ǝ}_{\text{person}}(\lambda y.\textbf{tdy}(\textbf{talked-to}(x)(y))); \mathrm{S}\end{array}}{\begin{array}{l}\lambda\varphi_1.\textsf{someone} \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \lambda x.\text{Ǝ}_{\text{person}}(\lambda y.\textbf{tdy}(\textbf{talked-to}(x)(y))); \mathrm{S}\upharpoonright\mathrm{NP}\end{array}}\upharpoonright\mathrm{I}^1$$

$$\dfrac{\begin{array}{l}\lambda\sigma.\sigma(\textsf{everyone});\\ \mathbf{V}_{\text{person}};\\ \mathrm{S}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\end{array} \qquad \begin{array}{l}\lambda\varphi_1.\textsf{someone} \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{today};\\ \lambda x.\text{Ǝ}_{\text{person}}(\lambda y.\textbf{tdy}(\textbf{talked-to}(x)(y))); \mathrm{S}\upharpoonright\mathrm{NP}\end{array}}{\begin{array}{l}\textsf{someone} \circ \textsf{talked} \circ \textsf{to} \circ \textsf{everyone} \circ \textsf{today};\\ \mathbf{V}_{\text{person}}(\lambda x.\text{Ǝ}_{\text{person}}(\lambda y.\textbf{tdy}(\textbf{talked-to}(x)(y)))); \mathrm{S}\end{array}}\upharpoonright\mathrm{E}$$

A quantifier has the ordinary GQ meaning (Ǝ_{person} and $\mathbf{V}_{\text{person}}$ abbreviate the terms $\lambda P.\exists x[\textbf{person}(x) \wedge P(x)]$ and $\lambda P.\forall x[\textbf{person}(x) \rightarrow P(x)]$, respectively), but its phonology is a function of type (**st**→**st**)→**st**. By abstracting over the position in which the quantifier 'lowers into' in an S via the Vertical Slash Introduction rule (35a), we can obtain an expression of type $\mathrm{S}\upharpoonright\mathrm{NP}$ (phonologically **st**→**st**) (①), which can be given as an argument to the quantifier. Then, by function application via $\upharpoonright$E (②), the subject quantifier *someone* semantically scopes over the sentence and lowers its phonology to the 'gap' position kept track of by λ-binding in phonology. The same process takes place for the object quantifier *everyone* to complete the derivation. The scopal relation between multiple quantifiers depends on the order of application of this hypothetical reasoning. The surface scope reading is obtained by switching the order of the hypothetical reasoning for the two quantifiers (which results in the same string of words, but with the opposite scope relation.

This formalization of quantifying-in by Oehrle (1994) has later been extended by Barker (2007) for more complex types of scope-taking phenomena known as 'parasitic scope' (a notion coined by Barker (2007) where, in transformational terms, some expression takes scope at LF by parasitizing on the scope created by a different scopal operator's LF movement—in versions of (TL)CG of the sort discussed here, this corresponds to double lambda-abstraction via the order-insensitive slash) in the analysis of symmetrical predicates. Empirical application of parasitic scope include 'respective' readings (Kubota & Levine 2014b), 'split scope' of negative quantifiers (Kubota & Levine 2016a) and modified numerals such as *exactly N* (Pollard 2014).

Hypothetical reasoning with prosodic λ-binding enables a simple analysis of *wh* extraction too, as originally noted by Muskens (2003). The key idea is that sentences with medial gaps can be analyzed as expressions of type $\mathrm{S}{\upharpoonright}\mathrm{NP}$, as in the derivation for (37) in (38).

(37) Bagels$_i$, Kim gave t_i to Chris.

(38)

$$
\dfrac{
\begin{array}{l}\textsf{bagels};\\ \mathbf{b};\mathrm{NP}\end{array}
\quad
\dfrac{
\begin{array}{l}\lambda\sigma\lambda\varphi.\varphi\circ\sigma(\epsilon);\\ \lambda\mathcal{F}.\mathcal{F};\\ (\mathrm{S}{\upharpoonright}X){\upharpoonright}(\mathrm{S}{\upharpoonright}X)\end{array}
\quad
\dfrac{
\dfrac{
\begin{array}{l}\textsf{kim};\\ \mathbf{k};\mathrm{NP}\end{array}
\quad
\dfrac{
\dfrac{
\begin{array}{l}\textsf{gave};\\ \mathbf{gave};\\ \mathrm{VP/PP/NP}\end{array}
\quad
\begin{bmatrix}\varphi;\\ x;\\ \mathrm{NP}\end{bmatrix}^1
}{\textsf{gave}\circ\varphi;\ \mathbf{gave}(x);\ \mathrm{VP/PP}}{/\mathrm{E}}
\quad
\begin{array}{l}\textsf{to}\circ\textsf{chris};\\ \mathbf{c};\mathrm{PP}\end{array}
}{\textsf{gave}\circ\varphi\circ\textsf{to}\circ\textsf{chris};\ \mathbf{gave}(x)(\mathbf{c});\ \mathrm{VP}}{/\mathrm{E}}
}{\textsf{kim}\circ\textsf{gave}\circ\varphi\circ\textsf{to}\circ\textsf{chris};\ \mathbf{gave}(x)(\mathbf{c})(\mathbf{k});\ \mathrm{S}}{\backslash\mathrm{E}}
}{\begin{array}{c}\lambda\varphi.\textsf{kim}\circ\textsf{gave}\circ\varphi\circ\textsf{to}\circ\textsf{chris};\\ \lambda x.\mathbf{gave}(x)(\mathbf{c})(\mathbf{k});\mathrm{S}{\upharpoonright}\mathrm{NP}\end{array}}\textcircled{1}
}{\lambda\varphi.\varphi\circ\textsf{kim}\circ\textsf{gave}\circ\textsf{to}\circ\textsf{chris};\ \lambda x.\mathbf{gave}(x)(\mathbf{c})(\mathbf{k});\ \mathrm{S}{\upharpoonright}\mathrm{NP}}\textcircled{2}
}{\textsf{bagels}\circ\textsf{kim}\circ\textsf{gave}\circ\textsf{to}\circ\textsf{chris};\ \mathbf{gave}(\mathbf{b})(\mathbf{c})(\mathbf{k});\ \mathrm{S}}{\upharpoonright}\mathrm{E}
$$

Here, after deriving an $\mathrm{S}{\upharpoonright}\mathrm{NP}$, which keeps track of the gap position via the λ-bound variable φ, the topicalization operator fills in the gap with an empty string and concatenates the topicalized NP to the left of the string thus obtained. This way, the difference between 'overt' and 'covert' movement reduces to a lexical difference in the prosodic specifications of the operators that induce them. Covert movement operator throws in some material in the gap position, whereas the overt movement operator 'closes off' the gap with an empty string.

As illustrated above, hypothetical reasoning for the Lambek slashes / and \ and for the vertical slash ↾ have important empirical motivations, but the real strength of a 'hybrid' system like Hybrid TLCG which recognizes both mechanisms is that it extends automatically to cases in which 'directional' and 'non-directional' phenomena interact. A case in point comes from the interaction of nonconstituent coordination and quantifier scope. Examples such as those in (39) allow for at least a reading in which the shared quantifier outscopes conjunction.[14]

[14] Whether the other scopal relation (one in which the quantifier meaning is 'distributed' to each conjunct, as in the paraphrase 'I gave a couple of books to Pat on Monday and to Sandy on Tuesday' for (39)) is possible seems to depend on various factors. With downward-entailing

(39) a. I gave a couple of books to Pat on Monday and to Sandy on Tuesday.

b. Terry said nothing to Robin on Thursday or to Leslie on Friday.

I now illustrate how this wide scope reading for the quantifier in NCC sentences like (39) is immediately predicted to be available in the fragment developed so far (Hybrid TLCG actually predicts both scopal relations for all NCC sentences; see Kubota & Levine (2015) for how the distributive scope is licensed). The derivation for (39b) is given in (40).

(40)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\varphi_1; P; \mathrm{VP/PP/NP}]^1 \quad [\varphi_2; x; \mathrm{NP}]^2}{\varphi_1 \circ \varphi_2; P(x); \mathrm{VP/PP}}/\mathrm{E} \quad \begin{array}{l}\textsf{to} \circ \textsf{robin};\\ \mathbf{r}; \mathrm{PP}\end{array}}{\varphi_1 \circ \varphi_2 \circ \textsf{to} \circ \textsf{robin}; P(x)(\mathbf{r}); \mathrm{VP}}/\mathrm{E} \quad \begin{array}{l}\textsf{on} \circ \textsf{thursday};\\ \mathbf{onTh}; \mathrm{VP\backslash VP}\end{array}}{\varphi_1 \circ \varphi_2 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday}; \mathbf{onTh}(P(x)(\mathbf{r})); \mathrm{VP}}\backslash\mathrm{E}}{\varphi_2 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday}; \lambda P.\mathbf{onTh}(P(x)(\mathbf{r})); \mathrm{(VP/PP/NP)\backslash VP}}\backslash\mathrm{I}^1}{\textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday}; \lambda x\lambda P.\mathbf{onTh}(P(x)(\mathbf{r})); \mathrm{NP\backslash(VP/PP/NP)\backslash VP}}\backslash\mathrm{I}^2}{}$$

$$\dfrac{\begin{array}{l}\vdots\\ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday};\\ \lambda x\lambda P.\mathbf{onTh}(P(x)(\mathbf{r}));\\ \mathrm{NP\backslash(VP/PP/NP)\backslash VP}\end{array} \quad \dfrac{\begin{array}{l}\textsf{or};\\ \lambda\mathcal{V}\lambda\mathcal{W}.\mathcal{W} \sqcup \mathcal{V};\\ (X\backslash X)/X\end{array} \quad \begin{array}{l}\vdots\\ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda x\lambda P.\mathbf{onFr}(P(x)(\mathbf{l}));\\ \mathrm{NP\backslash(VP/PP/NP)\backslash VP}\end{array}}{\begin{array}{l}\textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda\mathcal{W}.\mathcal{W} \sqcup [\lambda x\lambda P.\mathbf{onFr}(P(x)(\mathbf{l}))];\\ \mathrm{(NP\backslash(VP/PP/NP)\backslash VP)\backslash(NP\backslash(VP/PP/NP)\backslash VP)}\end{array}}/\mathrm{E}}{\begin{array}{l}\textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda x\lambda P\lambda z.\mathbf{onTh}(P(x)(\mathbf{r}))(z) \vee \mathbf{onFr}(P(x)(\mathbf{l}))(z); \mathrm{NP\backslash(VP/PP/NP)\backslash VP}\end{array}}\backslash\mathrm{E}$$

quantifiers such as (39b), this reading seems difficult to obtain without heavy contextualization and appropriate intonational cues. See Kubota & Levine (2015) for some discussion.

$$\dfrac{\left[\begin{array}{l}\varphi_3;\\ x;\mathrm{NP}\end{array}\right]^3 \qquad \begin{array}{l}\vdots\\ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda x \lambda P \lambda z.\mathbf{onTh}(P(x)(\mathbf{r}))(z) \vee \\ \quad \mathbf{onFr}(P(x)(\mathbf{l}))(z);\\ \mathrm{NP}\backslash(\mathrm{VP}/\mathrm{PP}/\mathrm{NP})\backslash\mathrm{VP}\end{array}}{\begin{array}{l}\varphi_3 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda P \lambda z.\mathbf{onTh}(P(x)(\mathbf{r}))(z) \vee \\ \quad \mathbf{onFr}(P(x)(\mathbf{l}))(z);\\ (\mathrm{VP}/\mathrm{PP}/\mathrm{NP})\backslash\mathrm{VP}\end{array}}\ \backslash\mathrm{E}$$

$$\dfrac{\begin{array}{l}\textsf{said};\\ \mathbf{said};\\ \mathrm{VP}/\mathrm{NP}/\mathrm{PP}\end{array} \qquad \begin{array}{l}\varphi_3 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda P \lambda z.\mathbf{onTh}(P(x)(\mathbf{r}))(z) \vee \\ \quad \mathbf{onFr}(P(x)(\mathbf{l}))(z);\\ (\mathrm{VP}/\mathrm{PP}/\mathrm{NP})\backslash\mathrm{VP}\end{array}}{\begin{array}{l}\textsf{said} \circ \varphi_3 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda z.\mathbf{onTh}(\mathbf{said}(x)(\mathbf{r}))(z) \vee \mathbf{onFr}(\mathbf{said}(x)(\mathbf{l}))(z); \mathrm{VP}\end{array}}\ \backslash\mathrm{E}$$

$$\dfrac{\begin{array}{l}\textsf{terry};\\ \mathbf{t};\mathrm{NP}\end{array} \qquad \begin{array}{l}\textsf{said} \circ \varphi_3 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda z.\mathbf{onTh}(\mathbf{said}(x)(\mathbf{r}))(z) \vee \mathbf{onFr}(\mathbf{said}(x)(\mathbf{l}))(z); \mathrm{VP}\end{array}}{\dfrac{\begin{array}{l}\textsf{terry} \circ \textsf{said} \circ \varphi_3 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \mathbf{onTh}(\mathbf{said}(x)(\mathbf{r}))(\mathbf{t})\mathbf{onFr}(\mathbf{said}(x)(\mathbf{l}))(\mathbf{t}); \mathrm{S}\end{array}}{\begin{array}{l}\lambda\varphi_3.\textsf{terry} \circ \textsf{said} \circ \varphi_3 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda x.\mathbf{onTh}(\mathbf{said}(x)(\mathbf{r}))(\mathbf{t}) \vee \mathbf{onFr}(\mathbf{said}(x)(\mathbf{l}))(\mathbf{t}); \mathrm{S}\upharpoonright\mathrm{NP}\end{array}}\ {\upharpoonright}\mathrm{I}^3}\ \backslash\mathrm{E}$$

$$\dfrac{\begin{array}{l}\lambda\sigma.\sigma(\textsf{nothing});\\ \neg\exists_{\mathrm{thing}};\\ \mathrm{S}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\end{array} \qquad \begin{array}{l}\lambda\varphi_3.\textsf{terry} \circ \textsf{said} \circ \varphi_3 \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \\ \quad \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \lambda x.\mathbf{onTh}(\mathbf{said}(x)(\mathbf{r}))(\mathbf{t}) \vee \mathbf{onFr}(\mathbf{said}(x)(\mathbf{l}))(\mathbf{t}); \mathrm{S}\upharpoonright\mathrm{NP}\end{array}}{\begin{array}{l}\textsf{terry} \circ \textsf{said} \circ \textsf{nothing} \circ \textsf{to} \circ \textsf{robin} \circ \textsf{on} \circ \textsf{thursday} \circ \textsf{or} \circ \textsf{to} \circ \textsf{leslie} \circ \textsf{on} \circ \textsf{friday};\\ \neg\exists_{\mathrm{thing}}(\lambda x.\mathbf{onTh}(\mathbf{said}(x)(\mathbf{r}))(\mathbf{t}) \vee \mathbf{onFr}(\mathbf{said}(x)(\mathbf{l}))(\mathbf{t})); \mathrm{S}\end{array}}\ {\upharpoonright}\mathrm{E}$$

The key point in this derivation is that, via hypothetical reasoning, the string *to Robin on Thursday or to Leslie on Friday* forms a syntactic constituent with a full-fledged meaning assigned to it in the usual way. Then the quantifier takes scope above this whole coordinate structure, yielding the non-distributive, quantifier wide-scope reading.

Licensing the correct scopal relation between the quantifier and conjunction in the analysis of NCC remains a challenging problem in the HPSG literature. See section 4.2.1 for some discussion.

2.5.3 Notes on other variants of TLCG

2.5.3.1 Displacement Calculus and NL$_\lambda$ Among different variant of TLCG, Morrill's Displacement Calculus and Barker and Shan's continuation-based calculus NL$_\lambda$ are most closely related to Hybrid TLCG. Roughly speaking, Hybrid TLCG's vertical slash $\upharpoonright$ plays more or less the same role as the discontinuity connectives $\uparrow$ and $\downarrow$ in Displacement Calculus and the 'continuation' slashes // and \\ in NL$_\lambda$. Many empirical analyses of linguistic phenomena formulated in one of these variants of TLCG translate to the other two more or less straightforwardly (for example, the analyses of Gapping and symmetrical predicates in Kubota & Levine (2016a) and Kubota & Levine (2016b), whose key ideas are briefly sketched above, build on Morrill's and Barker's analyses of the respective phenomena), though there are some important differences (some of which I point out below).

One major difference between the Displacement Calculus and Hybrid TLCG on the one hand and NL_λ on the other is that the latter takes **NL**, namely, the Non-associative Lambek Calculus, as the underlying calculus for the directional slashes / and \. Barker & Shan (2015) briefly comment on this property of their system, alluding to the possibility of controlling flexibility of constituency via the notion of 'structural control' (see section 2.5.3.2 below) in Multi-Modal Type-Logical Grammar. This certainly is a viable view, but no explicit extension of NL_λ along these lines currently exists. Morrill's approach differs from K&L's in certain important ways in the treatment of specific linguistic phenomena. The most substantial disagreement pertains to the treatment of island constraints. Unlike K&L, who take (most) island constraints to be processing-oriented, Morrill consistently holds the view that major island constraints should be treated within the narrow syntax (Morrill 1994; Morrill 2010; 2017). See also a brief discussion about determiner gapping in section 4.2.2 for another point of disagreement.

2.5.3.2 Multi-Modal TLCG TLCG come in different varieties, including Hybrid TLCG and the Displacement Calculus. There is a line of work most actively investigated back in the 90s and whose core architecture provides the theoretical underpinnings to Barker and Shan's NL_λ, called Multi-Modal Categorial Type Logics (MMCTL) (Moortgat & Oehrle 1994; Moortgat 2011; Bernardi 2002; Vermaat 2005) (but it should be kept in mind that the actual form that NL_λ takes as a linguistic theory and the kinds of linguistic phenomena most extensively studied in it differs considerably from MMCTL research from the 90s). One crucial difference between Hybrid TLCG and the Displacement Calculus on the one hand and MMCTL on the other is that, instead of recognizing a separate level of prosodic representation, MMCTL deals with various (somewhat heterogeneous) phenomena ranging from morpho-syntactic properties of verb clusters in Dutch cross-serial dependencies (Moortgat & Oehrle 1994) to technical difficulties with the Lambek Calculus in dealing with medial extraction (Moortgat 2011) (see the discussion in section X above) via the abstract notion of 'structural control', building on the technique of mixing different kinds of logic within a single deductive system originally developed in the literature of substructural logic (Restall 2000).

To see this point, it is instructive to take a look at the analysis of medial extraction in MMCTL, illustrated by the following derivation (adapted from Bernardi (2002)):

(41)

$$
\begin{prooftree}
\AxiomC{$(N\backslash N)/(S/_a NP) \vdash (N\backslash N)/(S/_a NP)$}
\AxiomC{$NP \vdash NP$}
\AxiomC{$(NP\backslash S)/_a NP \vdash (NP\backslash S)/_a NP$}
\AxiomC{$[\, NP \vdash NP \,]^1$}
\RightLabel{$/_a E$}
\BinaryInfC{$(NP\backslash S)/_a NP \circ_a NP \vdash NP\backslash S$}
\AxiomC{$(NP\backslash S)\backslash (NP\backslash S) \vdash (NP\backslash S)\backslash (NP\backslash S)$}
\RightLabel{$\backslash E$}
\BinaryInfC{$((NP\backslash S)/_a NP \circ_a NP) \circ (NP\backslash S)\backslash(NP\backslash S) \vdash NP\backslash S$}
\RightLabel{$\backslash E$}
\BinaryInfC{$NP \circ (((NP\backslash S)/_a NP \circ_a NP) \circ (NP\backslash S)\backslash(NP\backslash S)) \vdash S$}
\RightLabel{Diss}
\UnaryInfC{$NP \circ (((NP\backslash S)/_a NP \circ (NP\backslash S)\backslash(NP\backslash S)) \circ_a NP) \vdash S$}
\RightLabel{MixAssoc}
\UnaryInfC{$(NP \circ ((NP\backslash S)/_a NP \circ (NP\backslash S)\backslash(NP\backslash S))) \circ_a NP \vdash S$}
\RightLabel{$/_a I^1$}
\UnaryInfC{$NP \circ ((NP\backslash S)/_a NP \circ (NP\backslash S)\backslash(NP\backslash S)) \vdash S/_a NP$}
\RightLabel{$/E$}
\BinaryInfC{$(N\backslash N)/(S/_a NP) \circ ((NP \circ ((NP\backslash S)/_a NP \mathrm{O} (NP\backslash S)\backslash(NP\backslash S))) \circ_a NP) \vdash N\backslash N$}
\end{prooftree}
$$

Here, the derivation is written in the sequent style natural deduction (see Chapter 2), where a sequent is a construct of the form $\Gamma \vdash A$ where the consequent A is a syntactic type and the antecedent Γ is a structured object consisting of possibly multiple syntactic types that are combined with one another via binary connectives (of several sorts). On the leaves of the trees, we find axioms of the form $A \vdash A$ and the sequent on the final line of the proof tree is obtained by applying the rules of logic (such as the Elimination and Introduction rules similar to the ones we have posited above) and 'structural rules', which manipulate the forms of the antecedents of the sequents that are derived.

Intuitively, what is going on in the above derivation is that an object NP that is hypothesized on the right of a transitive verb is 'moved' to the right edge via the structural rule of Mixed Associativity in the structured antecedent on the third line from the bottom. This licenses the /aI rule, withdrawing the hypothesis to create the S/aNP category that is suitable as an argument to the relative pronoun. After lexical substitution, the derivation in (41) embodies, for example, a formal proof for the fact that the string 'which Sara wrote __ there' is a well-formed relative clause in English containing a medial gap.

There are a couple of important points of difference between MMCTL and other variants of TLCG that are already clear from this rather cursory exposition. First, the analysis of extraction via a series of structure rewriting operations in the abstract substructural component of the type logic is very different from the prosodic λ abstraction analysis presented above, due originally to Muskens (2003) and incorporated in many contemporary variants of TLCG (such as Hybrid TLCG and Pollard's Linear Categorial Grammar (Mihaliček & Pollard 2012; Pollard 2013)). Second, the exact ontological status of this 'multi-modal' substructural component is somewhat unclear. As noted above, the structure-changing operations have been put to use not just for extraction in English but for dealing with complex morpho-syntactic properties of verb clustering in Dutch cross-serial dependencies (Moortgat & Oehrle 1994). All this is done, not in a separate prosodic component, but inside a complex logic of syntactic types, where these syntactic types are taken as primitives that enter into binary composition operations of various sorts having different combinatorial possibilities. The different ontological setup in different variants of contemporary TLCG may have to do with the different research goals and research practices. When the emphasis is on linguistic application, a clear separation of ontologically distinct components seems important, but when the emphasis is on studying the meta-logical properties of the formal calculus, building directly on the rich literature of substructural logic and formalizing the type logic for natural language syntax literally *as* a substructural logic certain seems to be a natural choice.

2.5.3.3 Linear Categorial Grammar There is a family of related approaches in the TLCG tradition (Oehrle 1994; de Groote 2001; Muskens 2003; Mihaliček & Pollard 2012; Pollard 2013) that are distinctly different from the variants of TLCG discussed above in *not* recognizing the directional slashes of the Lambek calculus as primitive logical connectives. I call this family of approaches Linear Categorial Grammar (LCG) in what follows.[15]

LCG is essentially a subsystem of Hybrid TLCG without the rules for the directional slashes. The original conceptual motivation for this architecture is similar to the motivation for linearization-based HPSG—the idea, originally due to Curry (1961), that separating the combinatoric component of grammar and surface word order realization leads to a cleaner theoretical architecture. To get a flavor of LCG, an analysis of *wh*-extraction from section 2.5.2 can be reformulated in an LCG fragment as follows:

[15] Among these variants, Abstract Categorial Grammar (ACG) has mostly been studied in the formal grammar literature as a meta-framework for embedding different types of linguistic theories (as opposed to as a linguistic theory by itself), despite the fact that de Groote (2001), the very first paper that describes it, motivated it by pointing out some empirical issues with standard variants of TLCG. For a highly readable and interesting recent work on using ACG as a meta-framework for embedding Tree Adjoining Grammar, see Pogodalla (2017).

(42)

$$
\dfrac{
\begin{array}{l}\text{bagels};\\ \mathbf{b}; \mathrm{NP}\end{array}
\quad
\dfrac{
\begin{array}{l}\lambda\sigma\lambda\varphi.\\ \quad \varphi \circ \sigma(\epsilon);\\ \lambda\mathcal{F}.\mathcal{F};\\ (\mathrm{S}\upharpoonright X)\upharpoonright(\mathrm{S}\upharpoonright X)\end{array}
\quad
\dfrac{
\dfrac{
\begin{array}{l}\text{kim};\\ \mathbf{k};\\ \mathrm{NP}\end{array}
\quad
\dfrac{
\dfrac{
\begin{array}{l}\lambda\varphi_1\lambda\varphi_2\lambda\varphi_3.\\ \quad \varphi_3 \circ \text{gave} \circ \varphi_1 \circ \varphi_2;\\ \mathbf{gave};\\ \mathrm{S}\upharpoonright\mathrm{NP}\upharpoonright\mathrm{PP}\upharpoonright\mathrm{NP}\end{array}
\quad
\begin{bmatrix}\varphi;\\ x;\\ \mathrm{NP}\end{bmatrix}^1
}{
\begin{array}{l}\lambda\varphi_2\lambda\varphi_3.\varphi_3 \circ \text{gave} \circ \varphi \circ \varphi_2;\\ \mathbf{gave}(x); \mathrm{S}\upharpoonright\mathrm{NP}\upharpoonright\mathrm{PP}\end{array}
}{\upharpoonright\mathrm{E}}
\quad
\begin{array}{l}\text{to} \circ \text{chris};\\ \mathbf{c}; \mathrm{PP}\end{array}
}{
\lambda\varphi_3.\varphi_3 \circ \text{gave} \circ \varphi \circ \text{to} \circ \text{chris}; \mathbf{gave}(x)(\mathbf{c}); \mathrm{S}\upharpoonright\mathrm{NP}
}{\upharpoonright\mathrm{E}}
}{
\text{kim} \circ \text{gave} \circ \varphi \circ \text{to} \circ \text{chris}; \mathbf{gave}(x)(\mathbf{c})(\mathbf{k}); \mathrm{S}
}{\backslash\mathrm{E}}
}{
\lambda\varphi.\text{kim} \circ \text{gave} \circ \varphi \circ \text{to} \circ \text{chris}; \lambda x.\mathbf{gave}(x)(\mathbf{c})(\mathbf{k}); \mathrm{S}\upharpoonright\mathrm{NP}
}{\upharpoonright\mathrm{I}^1}
}{
\lambda\varphi.\varphi \circ \text{kim} \circ \text{gave} \circ \text{to} \circ \text{chris}; \lambda x.\mathbf{gave}(x)(\mathbf{c})(\mathbf{k}); \mathrm{S}\upharpoonright\mathrm{NP}
}{\upharpoonright\mathrm{E}}
}{
\text{bagels} \circ \text{kim} \circ \text{gave} \circ \text{to} \circ \text{chris}; \mathbf{gave}(\mathbf{b})(\mathbf{c})(\mathbf{k}); \mathrm{S}
}{\upharpoonright\mathrm{E}}
$$

The key difference from the Hybrid TLCG derivation above in (38) is that here the syntactic categories of linguistic expressions do not specify word order. Note in particular that the ditransitive verb *gave* is syntactically of type $\mathrm{S}\upharpoonright\mathrm{NP}\upharpoonright\mathrm{PP}\upharpoonright\mathrm{NP}$, and specifies the order of the arguments relative to the verb itself in the prosodic representation, which is formally a function of type $\mathbf{st}{\to}\mathbf{st}{\to}\mathbf{st}{\to}\mathbf{st}$.

While the clear separation of the underlying combinatorics and surface word order is certainly appealing, the conceptual elegance comes at some nontrivial empirical cost. Specifically, the simple analysis of NCC as constituent coordination, discussed in section 2.4.1, which is the hallmark of many variants of CG, does not straightforwardly carry over to this setup. This issue was already anticipated by Muskens (2001), and is discussed in detail in Kubota & Levine (2015). To see what's at issue, note that the distinction between S/NP and NP\S, which is crucial for distinguishing well-formed RNR sentence and ungrammatical strings in the CCG/Lambek calculus analysis of RNR is lost in LCG, since the two will be collapsed into the same type $\mathrm{S}\upharpoonright\mathrm{NP}$. But then, neither the syntactic type nor the prosodic type (which is uniformly $\mathbf{st}{\to}\mathbf{st}$) of the conjuncts provides enough information to determine conjoinability.

Worth (2016) presents the most comprehensive treatment of coordination in LCG addressing this problem (see also Kanazawa (2015)). The core idea of his proposal is to encode word order information via fine-grained subtypes of the prosodic types of linguistic expressions. Whether this extension of LCG offers the same (or better) empirical coverage as more traditional variants of TLCG (including Hybrid TLCG) is currently an open question.

[It may be useful to include a discussion of Minimalist Grammars in this section, but it may also be a bit tricky.]

3 Architectural similarities and differences

3.1 Broad architecture

One important property common to HPSG and CG is that they are both 'lexicalist' theories of syntax in the broader sense.[16] This is due to an explicit choice made at an early stage of the development of HPSG to encode valence information in the syntactic categories of linguistic expressions, following CG (see Chapters 2 and 4).[17] The two theories share many similarities in the analyses of specific linguistic phenomena due to this basic architectural similarity. For example, many phenomena that are treated by means of A movement operations (or via empty categories) in mainstream syntax, such as passivization, raising/control in English and 'complex predicate' phenomena in a typologically broad range of languages are generally treated by the sharing of valence information in the lexicon in these theories. For HPSG analyses of these phenomena, see Chapters 9 and 13. Steedman & Baldridge (2011) contains a good summary of CG analyses of local dependencies (passivization, raising/control). Kubota (2014) contains a comparison of HPSG and CG analyses of complex predicates. The heavy reliance on 'lexicalist' analyses of local dependencies is perhaps the most important property that is shared in common in HPSG and various versions of CG.

Need to check where the issue of 'lexical integrity hypothesis' is discussed in the book.

But emphasizing this commonality too much may be a bit misleading, since the valence features of HPSG and the slash connectives in CG have very different ontological statuses in the respective theories. The valence features in HPSG are primarily specifications, closely tied to the specific phrase structure rules, that specify the ways in which hierarchical representations are built. To be sure, the lexical specifications of the valence information play a key role in the movement-free analyses of local dependencies along the lines noted above, but still, there is a rather tight connection between these valence specifications originating in the

[16] I say 'broader sense' here since not all variants of either HPSG or CG subscribe to the so-called 'lexical integrity hypothesis', which says that syntax and morphology are distinct components of grammar. For example, the treatments of verb clustering in Dutch by Moortgat & Oehrle (1994) and in Japanese by Kubota (2014), seem to go against the tenet of the lexical integrity hypothesis. See also Chapter 4 for some discussion on lexicalism.

[17] This point is explicitly noted by the original founders of HPSG in the following passage in Pollard & Sag (1987: 11):

> A third principle of universal grammar posited by HPSG, the Subcategorization Principle, is essentially a generalization of the 'argument cancellation' employed in categorial grammar

lexicon and the ways in which they are 'cancelled' in specific phrase structure rules.

Things are quite different in CG, especially in TLCG. As discussed in section 2, TLCG views the grammar of natural language *not* as a structure-building system, but as a logical deductive system. The two slashes / and \ are thus not 'features' that encode the subcategorization properties of words in the lexicon, but have a much more general and fundamental role within the basic architecture of grammar in TLCG. These connectives are literally implicational connectives within a logical calculus. Thus, in TLCG, 'derived' rules such as type-raising and function composition are *theorems*, in just the same way that the transitivity inference is a theorem in classical propositional logic. Note that this is not just a matter of high-level conceptual organization of the theory, since, as discussed in section 2, the ability to assign 'constituent' statuses to non-canonical constituents in the CG analyses of NCC directly exploits this property of the underlying calculus. The straightforward mapping from syntax to semantics discussed in section 2.3 is also a direct consequence of adopting this 'derivation as proof' perspective on syntax, building on the results of Curry-Howard correspondence ***[ref]*** in setting up the syntax-semantics interface.[18]

Another notable difference between (especially a recent variant of) HPSG and CG is that CG currently lacks a detailed theory of 'constructions', that is, patterns and (sub)regularities that are exhibited by linguistic expressions that cannot (at least according to the proponents of 'constructionist' approaches) be lexicalized easily. As discussed in Chapter 37, recent Sign-Based Construction Grammar (SBCG) variants of HPSG (Sag et al. 2012) incorporates ideas from Construction Grammar (Goldberg 1995) and capture such generalizations via a set of constructional templates (or schemata), which are essentially a family of related phrase structure rules that are organized in a type inheritance hierarchy.

Such an architecture seems nearly impossible to implement literally in CG, except via of empty operators or lexical operations corresponding to each such constructional schema. In particular, in TLCG, syntactic rules are logical inference rules, so, there is no option to freely add syntactic rules in the deductive system. However, the crucial tenet of Construction Grammar in its strongest form, namely, the claim that the grammar of natural language robustly exhibits patterns that cannot be perspicuously captured within the lexicon seems still highly

[18] Although CCG does not embody the idea of 'derivation as proof' as explicitly as TLCG does, it remains true to a large extent that the role of the slash connective within the overall theory is different from that of the valence features in HPSG, given that CCG and TLCG share share many key ideas in the analyses of actual empirical phenomena.

controversial even within the the HPSG literature. Thus, if the relevant generalizations can all be lexicalized, along the lines argued by Mueller and Wechsler ***[ref]*** (see also Chapter 37 and Steedman & Baldridge (2011), the latter of which briefly discuss ways in which some of the empirical generalizations that Goldberg (1995) adduces to the notion of constructions can be lexicalized within CCG), then it may not pose any fundamental problem for the strongly lexicalist architecture of CG. This being said, it should be noted that much less research has been conducted on the so-called 'peripheral' phenomena from a CG perspective, and incorporating the recent advances on the analyses of these phenomena in the SBCG literature remains to be an important task for future CG research.

3.2 Syntax-semantics interface

As should be clear from the above exposition, both CCG and TLCG (at least in the simplest form) adopt a very rigid, one-to-one correspondence between syntax and semantics. Steedman's work on CCG has demonstrated that this simple and systematic mapping between syntax and semantics enables attractive analyses of a number of empirical phenomena at the syntax-semantics interface, including some notorious problems such as the scope parallelism issue in right-node raising known as Geach sentences (*Every boy loves, and every girl detests, some saxophonist*). Other important work on issues at the syntax-semantics interface include Jacobson's (1999; 2000) work on pronominal anaphora in Variable-Free Semantics (covering a wide range of phenomena including the paycheck/Bach-Peters paradigms and binding parallelism in right-node raising), Barker & Shan's (2015) work on 'continuation-based' semantics (weak crossover, superiority effects and 'parasitic scope' treatments of symmetrical predicates and sluicing) and Kubota and Levine's (2015; 2017) Hybrid TLCG, dealing with interactions between coordination, ellipsis and scopal phenomena.

As discussed in Chapter 24, recent HPSG work on complex empirical phenomena at the syntax-semantics interface makes heavy use of underspecification. For example, major analyses of nonconstituent coordination in recent HPSG use some version of underspecification framework to deal with complex interactions between coordination and scopal operators. (Beavers & Sag 2004; Yatabe 2001; Park et al. 2018). In a sense, HPSG retains a rigid phrase structure-based syntax (modulo the flexibility entertained with the use of the linearization-based architecture) and deals with the complex mapping to semantics via the use of underspecification languages in the semantic component such as MRS and LRS. CG, on the other hand, sticks to a tight mapping from syntax to semantics, but

need to check Chapter 24

makes the syntactic component itself flexible.[19] These differences may again be a reflection of a somewhat broader 'philosophical' difference between the two theories alluded to above (HPSG's surface-oriented syntax vs. CG's abstract view on the nature of the combinatoric component).

3.3 Morpho-syntax and word order

[This section is not yet written.]

need to check Chapter 23 to see what's discussed in there

Some things which seem worth discussing here:

- Linearization HPSG (Reape, Kathol) vs. multi-modal CG (Moortgat/Oehrle, Baldridge)
- treatment of features — more detailed development in HPSG as compared to CG (but see Morrill 1994)

  cf. Also:

  - Unification-based CG (Uszkoreit, Zeevat)
  - Bayer & Johnson (1995); Bayer (1996) vs. HPSG work on feature neutralization

need to check whether/ where feature neutralization is discussed in the book

- Recent advances in work on morphology in HPSG (Bonami/Crysmann) but no comparable detailed theory in CG

  But see Carpenter on lexical rules in CG; also the idea of categorial morphology in Hoeksema's early work and the paper by ???

4 Specific empirical phenomena

Since Part II of the present handbook contains an excellent introduction to recent developments of HPSG research on major linguistic phenomena, here I will try to highlight the differences between HPSG and CG in the analyses of a selected empirical phenomena. So as to make the ensuing discussion maximally informative, I choose to focus on phenomena over which there is some ongoing major

[19]But note that in principle this architecture does not preclude the use of underspecification. Steedman's (2012) recent approach to quantification crucially employs a limited use of underspecification for the treatment of indefinites. Similarly, Bekki and Mineshima's Dependent Type Semantics (Bekki 2014; Bekki & Mineshima 2017), a version of compositional proof theoretic semantics that has a straightforward interface to CG syntax, makes use of underspecification for the treatment of anaphora.

cross-theoretical debate, and those for which I believe one or the other theory would benefit from recent developments/rich research tradition in the other.

4.1 Long-distance dependencies

As noted in section 2.4, CCG treats long-distance dependencies via a sequence of function composition, which is similar to the SLASH percolation analysis in HPSG. CCG offers a treatment of major aspects of long-distance dependencies, including island effects (Steedman 2000) and parasitic gaps (Steedman 1987). Earlier versions of CCG involved a somewhat ad-hoc stipulation on the use of 'crossed composition' rules (Steedman 1997), but this was overcome in the more recent, 'multi-modal' variant of CCG (Baldridge 2002), which controls the application of such non-order-preserving rules via a fine-grained system of 'lexicalized modality' by means of the lexical specifications of the relevant linguistic expressions.

The situation is somewhat different in TLCG. TLCG typically makes use of a movement-like operation for the treatment of extraction phenomena, but the specific implementations differ considerably in different variants of TLCG. Major alternatives include the notion of 'structural control' in multi-modal variants of TLCG (Morrill 1994; Moortgat 2011), and prosodic λ-binding in LCG and related approaches (see section 2.5.2). In either approach, extraction phenomena are treated by means of some form of hypothetical reasoning, and this raises a major technical issue. The underlying calculus of TLCG is a version of linear logic, and this means that the implication connective is resource sensitive. This is problematic in situations in which a single filler corresponds to multiple gaps, as in parasitic gaps and related phenomena. These cases of extraction require some sort of extension of the underlying logic or some special operator that is responsible for resource duplication. Currently, the most detailed treatment of extraction phenomena in the TLCG literature is Morrill (2017), which lays out in detail an analysis of long-distance dependencies capturing both major island constraints and parasitic gaps within the most recent version of Morrill's Displacement Calculus.

There are several complex issues that arise in relation to the linguistic analysis of extraction phenomena. One major open question is whether island constraints should be accounted for within narrow grammar. Both Steedman and Morrill follow the standard practice in generative grammar research in taking island effects to be syntactic, but this consensus has been challenged by a new body of research in the recent literature proposing various alternative explanations on different types of island constraints (see Chapter 16, Levine (2017) and Newmeyer

(2016) for an overview of this line of work). Recent syntactic analyses of long-distance dependencies in the HPSG literature explicitly avoid directly encoding major island constraints within the grammar (Sag 2010; Chaves 2012b).

Another major empirical problem in connection to the analysis of long-distance dependencies is the so-called extraction pathway marking phenomenon (McCloskey 1979; Zaenen 1983). While this issue has received considerable attention in the HPSG literature, through a series of work by Levine and Hukari (see Levine & Hukari 2006), there is currently no explicit treatment of this phenomenon in the CG literature. CCG can probably incorporate the HPSG analysis relatively easily, given the close similarity of the basic analysis of extraction in the two approaches, which crucially involves keeping track of extraction pathway explicitly in the syntactic category of the expression containing a gap at each step of syntactic composition (but no explicit analysis has been worked out to date). Extraction pathway marking seems to pose a somewhat trickier challenge to analyses of extraction in TLCG, which typically do not involve the HPSG/CCG type step-by-step explicit encoding of extraction pathways. How to address this issue in TLCG is currently one major empirical question in TLCG research.

Is this (going to be) discussed in Chapter 14?

4.2 Coordination and ellipsis

Coordination and ellipsis are both major issues in contemporary syntactic theory. There are moreover some phenomena, such as Gapping and Stripping, which seem to lie at the boundary between the two empirical domains. There are some important similarities and differences between analytic ideas entertained in the HPSG and CG literature for problems in these empirical domains, which I will try to highlight in the following discussion.

4.2.1 Analyses of nonconstituent coordination

CG is perhaps best known in the wider linguistic literature for its analysis of nonconstituent coordination. Steedman's work on CCG (Steedman 1997; 2000; Steedman 2012) in particular has shown how this analysis of coordination interacts smoothly with analyses of other major linguistic phenomena (such as long-distance dependencies, control and raising and quantification) to achieve a surface-oriented grammar that has wide empirical coverage and at the same time has attractive computational properties. Kubota & Levine (2015) offer an up-to-date TLCG analysis of coordination, and compare it with major alternatives in both CCG and HPSG literature.

As compared to long-distance dependencies, coordination (in particular NCC) has received considerably less attention in the (H)PSG literature initially (Sag et al. (1985) is an important exception in the early literature). Things started to change somewhat around 2000, with a series of related proposals including Yatabe (2001), Crysmann (2008), Beavers & Sag (2004) and Chaves (2007) (see Chapters 17 and 20). Here, I take up Beavers & Sag (2004) (B&S) and Yatabe (2001) as two representative proposals in this line of work. The two proposals share some common assumptions and ideas, but they also differ in important respects which do not seem to have received enough attention in the literature (see also Yatabe & Tam (2017) for a more detailed comparison).

Both B&S and Yatabe (2001) adopt linearization-based HPSG, together with (a version of) Minimal Recursion Semantics for semantics. Of the two, B&S's analysis is more in line with standard assumptions in HPSG. The basic idea of B&S's analysis is indeed very simple: by exploiting the flexible mapping between the combinatoric component and the surface word order realization in linearization-based HPSG, they essentially propose a surface deletion-based analysis of NCC according to which NCC examples are analyzed as follows:

need to adjust the discussion here on the basis of what's in the ellipsis chapter (already in place)

(43) [S Terry gave no man a book on Friday] or [S ~~Terry gave no man~~ a record on Saturday].

where the material in strike-out is underlyingly present but undergoes deletion in the prosodic representation.

In its simplest form, this analysis gets the scopal relation between the quantifier and coordination wrong in examples like (43) (a well-known problem for the conjunction reduction analysis from the 70s; cf. Partee (1970)). B&S address this issue by introducing a condition called 'Optional Quantifier Merger', which says the following:

(44) **Optional Quantifier Merger**: For any elided phrase denoting a generalized quantifier in the domain of either conjunct, the semantics of that phrase may optionally be identified with the semantics of its non-elided counterpart.

As noted by Levine (2011) and Kubota & Levine (2015), this condition does not follow from any general principle and is merely stipulated in B&S's account.

Yatabe (2001) and Yatabe & Tam (2017) (the latter of which contains a much more accessible exposition of essentially the same proposal as the former) propose a somewhat different analysis. Unlike B&S, who assume that semantic composition is carried out on the basis of the meanings of *signs* on each node (which is the standard assumption about semantic composition in HPSG), Yatabe shifts the locus of semantic composition to the list of domain objects, that is, the component that directly gets affected by the deletion operation that yields the surface string.

This crucially changes the default meaning predicted for examples such as (43). Specifically, on Yatabe's analysis, surface string for (43) is obtained by the 'compaction' operation on word order domains that collapses two quantifiers originally contained in the two conjuncts into one. The semantics of the whole sentence is computed on the basis of this resultant word order domain representation, which contains only *one* instance of a domain object corresponding to the quantifier. The quantifier is then required to scope over the whole coordinate structure due to independently motivated principles of underspecification resolution. While this approach successfully yields the wide-scope reading for quantifiers, the distributive, narrow scope reading for quantifiers (which was trivial for B&S) now becomes a challenge. Yatabe and Tam simply stipulate a complex disjunctive constraint on semantic interpretation tied to the 'compaction' operation that takes place in coordination so as to generate the two scopal readings.

Not sure how much detail I should go into re the Y&T unpublished MS. This is important work (and different from B&S in some nontrivial respects), but at the same time, it's still unpublished.

Kubota & Levine (2015) note that B&S's approach suffers from similar issues in the interpretations of symmetrical predicates, summative predicates and 'respective' readings (see Chaves (2012a) for a lucid discussion of the empirical parallels between the three phenomena and how the basic cases can receive a uniform analysis within HPSG). Yatabe & Tam (2017) offer a response to K&L, arguing that Yatabe's (2001) analysis does not suffer from the problems that K&L point out, and discuss some empirical issues which they argue are problematic for CG analyses of coordination (see also Kubota & Levine (2018) for a rebuttal). At least part of the controversy (or confusion) in this debate seems to be related to different perspectives on what counts as linguistic analysis: is the primary goal of linguistic analysis an accurate description of observed facts, or do we want to aim for a deeper 'explanation'? There is no easy answer to this questions, and different research communities seem to have varying degrees of emphasis on the opposite poles of the spectrum, as the comparison of HPSG and minimalism in Chapter 33 makes clear.

4.2.2 Gapping and Stripping

Descriptively, Gapping is a type of ellipsis phenomenon that occurs in coordination and which deletes some material including the main verb:[20]

(45) a. Leslie **bought** a CD, and Robin ∅ a book.
b. Terry **can go** with me, and Pat ∅ with you.
c. John **wants to try to begin to write** a novel, and Mary ∅ a play.

Gapping has invoked some theoretical controversy in the recent HPSG/CG literature for the 'scope anomaly' issue that it exhibits. The relevant data involving auxiliary verbs such as (46a) and (46b) have long been known in the literature since Oehrle (1971; 1987), Siegel (1987) and McCawley (1993) later pointed out similar examples involving downward-entailing determiners such as (46c).

(46) a. Mrs. J can't live in Boston and Mr. J ∅ in LA.
b. Kim didn't play bingo or Sandy ∅ sit at home all evening.
c. No dog eats Whiskas or ∅ cat ∅ Alpo.

Kubota & Levine (2014a; 2016a) noting some difficulties for earlier accounts of Gapping in the (H)PSG literature and argue for a constituent coordination analysis of Gapping in TLCG, building on earlier analyses of Gapping in CG (Steedman 1990; Hendriks 1995b; Morrill & Solias 1993). The key idea of K&L's analysis involves taking Gapping as coordination of clauses missing a verb in the middle, which can be transparently represented as a function from strings to strings:

(47) $\lambda\varphi.\mathsf{leslie} \circ \varphi \circ \mathsf{a} \circ \mathsf{cd}$

A special type of conjunction entry then conjoins two such expressions and returns a conjoined sentence missing the verb only in the first conjunct (on the prosodic representation). By feeling the verb to this resultant expression, a proper form-meaning pair is obtained for Gapping sentences like those in (45).

The apparently unexpected wide scope readings for auxiliaries and quantifiers in (46) turns out to be straightforward on this analysis. I refer the interested

[20] There is some disagreement as to whether Gapping is restricted to coordination. Kubota & Levine (2016a), following authors such as Johnson (2009), take Gapping to be restricted to coordination. Park et al. (2018) take a different view, and argue that Gapping should be viewed as a type of ellipsis phenomenon that is not restricted to coordination environments.

reader to Kubota & Levine (2016a) for details, but the key idea is that the apparently 'anomalous' scope in such examples isn't really anomalous on this approach, since the auxiliary (which prosodically lowers into the first conjunct) takes the whole conjoined gapped clause as its argument. Thus, the existence of the wide scope reading is automatically predicted on this analysis. Puthawala (2018) extends this approach to a similar 'scope anomaly' data found in Stripping, another type of (coordinate) ellipsis phenomenon.

The determiner gapping example in (46c) requires a somewhat more elaborate treatment. Kubota & Levine (2016a) analyze determiner gapping via higher-order functions. Morrill & Valentín (2017) criticize this approach for a certain type of overgeneration problem regarding word order and propose an alternative analysis in Displacement Calculus.

Park et al. (2018) propose an analysis of Gapping in HPSG that overcomes the limitations of previous (H)PSG analyses of Gapping (Sag et al. 1985; Abeillé et al. 2014; Chaves 2005), couched in Lexical Resources Semantics. In their analysis, the lexical entries of the clause-level conjunction words *and* and *or* are underspecified as to the relative scope between the propositional operator contributed by the modal auxiliary in the first conjunct and the boolean conjunction or disjunction connective that is contributed by the conjunction word itself. Park et al. argue that this is sufficient to capture the scope anomaly in the Oehrle/Siegel data such as (46a) and (46b). Extension to the determiner gapping case (46c) is left for future work.

Need to check the status of the Park et al. paper. Is a written version already available?

At a somewhat general level, the main difference between the HPSG and (Hybrid) TLCG analyses of Gapping again seems to at least partly pertain to the level of linguistic analysis that the respective approaches mainly target. K&L's analysis crucially makes use of higher-order functions both in the semantic and prosodic components. This may pose issues, for example, in computational implementation or in the context of building an explicit human sentence processing model. The standard view in generative grammar of course is to separate the competence grammar and performance, but still, it seems legitimate to note that in the type of analysis advocated by K&L, the theoretical elegance is gained at the cost of making the relationship between the competence grammar and performance less direct. Park et al.'s (2018) approach on the other hand is more in line with the usual practice of HPSG research where the main emphasis is on writing an explicit grammar fragment that is constraint-based and surface-oriented. Again, this type of conflict is not easy to overcome, especially when (as in the present situation) the competing analyses are formulated in different theories. Perhaps we will arrive at a synthesizing view at some point in the future. Until then, it seems most fruitful for different approaches to pursue their own goals, while keeping an eye on alternative perspectives.

4.2.3 Ellipsis

Analyses of major ellipsis phenomena in HPSG and CG share the same essential idea that ellipsis is a form of anaphora, without any invisible hierarchically structured representations corresponding to the 'elided' expression. See Chapter 20 for an overview of approaches to ellipsis phenomena in HPSG. In this section, I try to highlight some possible differences between some of the representative proposals in the recent literature on ellipsis in HPSG and CG.

check ellipsis chapter (now complete) to see how much to include here

Recent analyses of ellipsis in HPSG (Ginzburg & Sag 2000; Miller 2014) make heavy use of the notion of 'constructions' adopted from Construction Grammar (this idea is even borrowed into some of the CG analyses of ellipsis such as Jacobson (2016)). Many ellipsis phenomena are known to exhibit some form of 'syntactic sensitivity' (Kennedy 2003; Chung 2013; Yoshida et al. 2015), and this fact has long been taken to provide strong evidence for the 'covert structure' analyses of ellipsis popular in mainstream syntactic literature (Merchant 2013). ***[Perhaps explain GS's analysis of sluicing a bit here and make it clear how it captures syntactic sensitivity]***

Some of the early work on ellipsis in CG include Hendriks (1995a) and Morrill & Merenciano (1996). Morrill & Merenciano (1996) in particular show how hypothetical reasoning in TLCG allows treatments of important properties of ellipsis phenomena such as strict/sloppy ambiguity and scope ambiguity of elided quantifiers in VP ellipsis. Jäger (2005) integrates these earlier works with a general theory of anaphora in TLCG incorporating the key empirical analyses of pronominal anaphora by Jacobson (1999; 2000). Jacobson's 1998; 2008 analysis of Antecedent-Contained Ellipsis is also important. ACE is often taken to offer a strong piece of evidence for the representational analysis of ellipsis in mainstream generative syntax. Jacobson offers a counterproposal to this standard analysis that completely dispenses with covert structural representations. While the above works from the 90s have mostly focused on VP ellipsis, recent developments in the CG literature, including Barker (2013) on sluicing, Jacobson (2016) on fragment answers and Kubota & Levine (2017) on pseudogapping, considerably extended the empirical coverage of the same line of analysis.

The relationship between recent CG analyses of ellipsis phenomena and HPSG counterparts seems to be similar to the situation with competing analyses on coordination. Both Barker (2013) and Kubota & Levine (2017) exploit hypothetical reasoning to treat the antecedent of an elided material as a 'constituent' with full-fledged semantic interpretation at an abstract combinatoric component of syntax. The anaphoric mechanism can then refer to both the syntactic and semantic information of the antecedent expression to capture syntactic sensitivity observed in ellipsis phenomena, without the need to posit hierarchical representations at the ellipsis site. Due to its surface-oriented nature, HPSG is not equipped with an analogous abstract combinatoric component that assigns 'constituent' statuses to expressions that do not (in any obvious sense) correspond to constituents in the surface representation. In HPSG, the major work in constraining the appropriate pairing of meaning and form is instead taken over by constructional schemata, which can both specify fine-grained details of specific subtypes of ellipsis phenomena and at the same time capture broader generalizations about ellipsis phenomena in general via highly articulate inheritance hierarchy organizing constructional schemata. This again pertains to a fundamental difference between the two theoretical frameworks, and at this point, it is unclear whether any consensus or synthesis of perspectives will ultimately emerge in the future.

4.2.4 Mismatches in right-node raising

- Some important work in the HPSG literature: Shiraishi et al. (EISS), Chaves's (2014) detailed analysis of RNR
- So far, no explicit analyses in the CG literature, though there does not seem to be any fundamental obstacle to treating some cases of RNR as instances of ellipsis within CG.

4.3 Binding

Empirical phenomena that have traditionally been analyzed by means of Binding Theory (both in the transformational and the nontransformational literature) potentially pose a major challenge to the 'non-representational' view of the syntax-semantics interface common to most variants of CG. As discussed in Chapter II, the HPSG Binding Theory captures Principle A and B effects at the level of argument structure, while Principle C makes reference to the configurational structure (i.e. the feature-structure encoding of the constituent geometry). The status of Principle C itself is controversial to begin with, but if this condition would need to be stated in the syntax, then, it would constitute the greatest challenge to CG-based theories of syntax.

While there seems to be no consensus in the current CG literature on how the standard binding theoretic facts are to be captured, there are some important ideas and proposals in the wider literature of CG-based syntax (broadly construed to include work in the Montague Grammar tradition). First, as for Principle A, there is a recurrent suggestion in the literature that these effects can (and should) be captured simply via strictly lexical properties of reflexive pronouns (e.g. Szabolcsi (1992); see Büring (2005) for a concise summary). For example, for a reflexive in the direct object position of a transitive verb bound by the subject NP, the following type assignment (where the reflexive pronoun first takes a transitive verb and then the subject NP as arguments) suffices to capture its bound status:

(48) himself; $\lambda R \lambda x . R(x)(x)$; NP\((NP\S)/NP)\S

While this approach is attractively simple, there are at least two things to keep in mind, in order to make it a complete analysis of Principle A within some contemporary variant of CG. First, while the lexical treatment of reflexive binding might at first sight (48) appear to capture the locality of binding quite nicely, CG's flexible syntax potentially overgenerates unacceptable long-distance binding readings for (English) reflexives. Since right-node raising can take place across clause boundaries, it seems necessary to assume that hypothetical reasoning for the Lambek-slash (or a chain of function composition that has the same effect in CCG) can generally take place across clause boundaries. But then, expressions such as *thinks Bill hates* can be assigned the same syntactic type (i.e. (NP\S)/NP) as lexical transitive verbs, overgenerating non-local binding of a reflexive from a subject NP in the upstairs clause (**John*$_i$ *thinks Bill hates himself*$_i$).

In order to prevent this situation while still retaining the 'reflexive as a local higher-order function' analysis sketched above, some kind of restriction needs to be imposed as to the way in which reflexives combine with other linguistic expressions (one possibility would be to incorporate some version of the notion of 'modal control' from the literature of Multi-Modal TLCG, to draw a distinction between lexical transitive verbs and derived transitive verb-like expressions by positing different 'modes of composition' in the two cases and making the lexical specification of the reflexive pronoun sensitive to this distinction).

The other issue is that the lexical entry in (48) needs to be generalized to cover all cases in which a reflexive is bound by an argument that is higher in the obliqueness hierarchy. This amounts to positing a polymorphic lexical entry for the reflexive. While the use of polymorphism is needed anyway in the grammar (at least in CG—e.g., to achieve a tight syntax-semantics correspondence in the analysis of coordination), unlike in standard accounts of binding phenomena in other lexicalist theories such as HPSG and LFG, this account would capture the Principle A effects purely via the specific lexical encoding for reflexive pronouns. In this sense, it is worth noting that such an account would not be a simple 'translational equivalent' of the HPSG binding theory, even though both are certainly 'lexicalist' analyses of binding effects in the broader sense.

While Principle A effects are in principle amenable to a quite straightforward lexical treatment, Principle B turns out to be considerably more challenging. To see this point, note that the lexical analysis of reflexives sketched above crucially relies on the fact that the constraint associated with reflexives corresponds to a straightforward semantic effect of variable binding. Pronouns instead require *disjointness* of reference from less oblique co-arguments, but such an effect cannot be captured by simply specifying some appropriate lambda term as the semantic translation for the pronoun.

To date, the most detailed treatment of Principle B effects in CG that explicitly addresses this difficulty is the proposal by Jacobson (2007), formulated in a version of CCG (standard CCG has its own treatment of syntactic Binding Principles, which effectively recognizes an intermediate level of representation for the purpose of stating Binding Principles—I will discuss this approach briefly at the end of this section).

[section incomplete; explain Jacobson's approach in what follows]
Other things to discuss:

- Steedman's take on binding
- What about Principle C?

5 Brief notes on proccessing and implementation

The discussion above has mostly focused on linguistic analysis. In this final section, I will briefly comment on CG's implications for psycholinguistics and computational linguistics research.

One attractive feature of CCG (but not CG in general), when viewed as an integrated model of the competence grammar and human sentence processing, is that it enables 'surface-oriented', incremental analyses of strings from left to right. This aspect was emphasized in the early literature of CCG (Ades & Steedman 1982; Crain & Steedman 1985), but it does not seem to have had much impact on psycholinguistic research in general since then. A notable exception is the work by Pickering & Barry (1991; 1993) in early 90s. There is some work on the relationship between processing and TLCG (see Morrill (2010: Chapters 9 and 10), and references therein). In any event, serious investigations of the relationship between competence grammar and human sentence processing from a CG perspective (either CCG or TLCG) seems to be a large untouched opportunity for future research, much like the situation with HPSG (see Chapter 28).

As for connections to computational linguistics/NLP research, like HPSG, large-scale computational implementation has been an important research agenda for CCG (see, e.g., Clark & Curran (2007)). I refer the reader to Steedman (2012: Chapter 13) for an excellent summary on this subject (this chapter contains a discussion of human sentence processing as well). Together with work on linguistically informed 'deep parsing' in HPSG, CCG parsers seem to be attracting some renewed interest in CL/NLP research recently, due to the new trend of combining the insights of statistical approaches and linguistically-informed approaches. In particular, the straightforward syntax-semantics interface of (C)CG seems to be an attractive feature in building CL/NLP systems that have an explicit compositional semantics component. See Lewis & Steedman (2013) and Mineshima et al. (2016) for this type of work. TLCG research has traditionally been less directly related to CL/NLP research. But there are recent attempts at constructing large-scale treebanks (Moot 2015) and combining TLCG frameworks with more mainstream approaches in NLP research such as distributional semantics (Moot 2017).

6 Conclusion

Abbreviations

Acknowledgements

References

Abeillé, Anne, Gabriela Bîlbîie & Francois Mouret. 2014. A Romance perspective on Gapping constructions. In H. Boas & F. Gonzálvez García (eds.), *Romance in construction grammar*, 227–265. Amsterdam: John Benjamins.

Ades, Anthony E. & Mark J. Steedman. 1982. On the order of words. *Linguistics and Philosophy* 4(4). 517–558.

Ajdukiewicz, Kazimierz. 1935. Die syntaktische Konnexität. *Studia Philosophica* 1. 1–27.

Baldridge, Jason. 2002. *Lexically specified derivational control in Combinatory Categorial Grammar*. University of Edinburgh dissertation.

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetic notation for syntactic descriptions. *Language* 29. 47–58.

Barker, Chris. 2007. Parasitic scope. *Linguistics and Philosophy* 30(4). 407–444.

Barker, Chris. 2013. Scopability and sluicing. *Linguistics and Philosophy* 36(3). 187–223.

Barker, Chris & Chung-chieh Shan. 2015. *Continuations and natural language*. Oxford: Oxford University Press.

Bayer, Samuel. 1996. The coordination of unlike categories. *Language* 72(3). 579–616.

Bayer, Samuel & Mark Johnson. 1995. Features and agreement. In Hans Uszkoreit (ed.), *33rd Annual Meeting of the Association for Computational Linguistics. Proceedings of the conference*, 70–76. Cambridge, MA: Association for Computational Linguistics.

Beavers, John & Ivan A. Sag. 2004. Coordinate ellipsis and apparent non-constituent coordination. In Stefan Müller (ed.), *Proceedings of the 11th International Conference on Head-Driven Phrase Structure Grammar, Center for Computational Linguistics, Katholieke Universiteit Leuven*, 48–69. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2004/, accessed 2018-2-25.

Bekki, Daisuke. 2014. Representing anaphora with dependent types. In Nicholas Asher & Sergei Soloviev (eds.), *Logical Aspects of Computational Linguistics 2014*, 14–29. Heidelberg: Springer.

Bekki, Daisuke & Koji Mineshima. 2017. Context-passing and underspecification in dependent type semantics. In Stergios Chatzikyriakidis & Zhaohui Luo (eds.), *Modern perspectives in type-theoretical semantics*, 11–41. Heidelberg: Springer.

Bernardi, Raffaella. 2002. *Reasoning with polarity in Categorial Type Logic*. University of Utrecht dissertation.

Borsley, Robert D. & Kersti Börjars (eds.). 2011. *Non-transformational syntax: Formal and explicit models of grammar: A guide to current models*. Oxford, UK/ Cambridge, MA: Blackwell Publishers Ltd.

Büring, Daniel. 2005. *Binding theory*. Cambridge, England: Cambridge University Press.

Carpenter, Bob. 1998. *Type-logical semantics*. Cambridge, MA: MIT Press.

Chaves, Rui. 2012a. Conjunction, cumulation and respectively readings. *Journal of Linguistics* 48(2). 297–344.

Chaves, Rui. 2012b. On the grammar of extraction and coordination. *Natural Language and Linguistic Theory* 30(2). 465–512.

Chaves, Rui P. 2014. On the disunity of right-node raising phenomena: Extraposition, ellipsis, and deletion. *Language* 4. 834–886. DOI:10.1353/lan.2014.0081

Chaves, Rui Pedro. 2005. A linearization-based approach to gapping. In G. Jäger, P. Monachesi, G. Penn & S. Wintner (eds.), *FG-MOL 2005: the 10th conference* *on Formal Grammar and the 9th meeting on mathematics of language*, 207–220. University of Edinburgh, Scotland.

Chaves, Rui Pedro. 2007. *Coordinate structures - constraint-based syntax-semantics processing*. Portugal: University of Lisbon dissertation. http://www.clul.ul.pt/clg/diss_rpc.pdf.

Chung, Sandra. 2013. Syntactic identity in sluicing: how much, and why. *Linguistic Inquiry* 44(1). 1–44.

Clark, Stephen & James Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics* 33(4). 493–552.

Crain, Stephen & Mark J. Steedman. 1985. On not being led up the garden path: The use of context by the psychological syntax processor. In David R. Dowty, Lauri Karttunen & Arnold M. Zwicky (eds.), *Natural language processing*, 320–358. Cambridge, UK: Cambridge University Press.

Crysmann, Berthold. 2008. An asymmetric theory of peripheral sharing in HPSG: Conjunction reduction and coordination of unlikes. In Gerhard Jäger, Paola Monachesi, Gerald Penn & Shuly Wintner (eds.), *Proceedings of Formal Grammar 2003, Vienna, Austria*, 47–62. Stanford, CA: CSLI Publications.

Curry, Haskell B. 1961. Some logical aspects of grammatical structure. In Roman Jakobson (ed.), *Structure of language and its mathematical aspects*, vol. 12 (Symposia on Applied Mathematics), 56–68. Providence: American Mathematical Society.

de Groote, Philippe. 2001. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter*, 148–155.

Dowty, David R. 1988. Type raising, functional composition, and nonconstituent coordination. In Richard Oehrle, Emmon Bach & Deirdre Wheeler (eds.), *Categorial Grammars and natural language structures*, 153–198. Dordrecht: D. Reidel Publishing Company.

Ginzburg, Jonathan & Ivan A. Sag. 2000. *Interrogative investigations: The form, meaning, and use of English interrogatives* (CSLI Lecture Notes 123). Stanford, CA: CSLI Publications.

Goldberg, Adele E. 1995. *Constructions: A Construction Grammar approach to argument structure* (Cognitive Theory of Language and Culture). Chicago/London: The University of Chicago Press.

Hendriks, Petra. 1995a. *Comparatives and categorial grammar*. University of Groningen dissertation.

Hendriks, Petra. 1995b. Ellipsis and multimodal categorial type logic. In Glyn V. Morrill & Richard T. Oehrle (eds.), *Formal Grammar: Proceedings of the Con-*

ference of the European Summer School in Logic, Language and Information, Barcelona, 1995, 107–122.

Jacobson, Pauline. 1998. Antecedent contained deletion and pied-piping: evidence for a variable-free semantics. In Devon Strolovitch & Aaron Lawson (eds.), *Proceedings from semantics and linguistic theory VIII*, 74–91. Ithaca, New York: Cornell University.

Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22(2). 117–184.

Jacobson, Pauline. 2000. Paycheck pronouns, Bach-Peters sentences, and variable=free semantics. *Natural Language Semantics* 8(2). 77–155.

Jacobson, Pauline. 2007. Direct compositionality and variable-free semantics: the case of "Principle B" effects. In Chris Barker & Pauline Jacobson (eds.), *Direct compositionality*, 191–236. Oxford: Oxford University Press.

Jacobson, Pauline. 2008. Direct compositionality and variable-free semantics: the case of Antecedent Contained Deletion. In Kyle Johnson (ed.), *Topics in ellipsis*, 30–68. Cambridge, England: Cambridge University Press.

Jacobson, Pauline. 2016. The short answer: implications for direct compositionality (and vice versa). *Language* 92(2). 331–375.

Jäger, Gerhard. 2005. *Anaphora and type-logical grammar*. Berlin: Springer.

Johnson, Kyle. 2009. Gapping is not (VP-) ellipsis. *Linguistic Inquiry* 40(2). 289–328.

Kanazawa, Makoto. 2015. Syntactic features for regular constraints and an approximation of directional slashes in Abstract Categorial Grammars. In Yusuke Kubota & Robert Levine (eds.), *Proceedings for ESSLLI 2015 Workshop 'Empirical Advances in Categorial Grammar'*, 34–70. University of Tsukuba & Ohio State University.

Kennedy, Christopher. 2003. Ellipsis and syntactic representation. In Kerstin Schwabe & Susanne Winkler (eds.), *The interfaces: deriving and interpreting omitted structures*, 29–53. Amsterdam: John Benjamins.

Kubota, Yusuke. 2010. *(In)flexibility of constituency in Japanese in Multi-Modal Categorial Grammar with Structured Phonology*. Ohio State University dissertation.

Kubota, Yusuke. 2014. The logic of complex predicates. English. *Natural Language and Linguistic Theory* 32(4). 1145–1204. DOI:10.1007/s11049-014-9246-8

Kubota, Yusuke. 2015. Nonconstituent coordination in Japanese as constituent coordination: an analysis in Hybrid Type-Logical Categorial Grammar. *Linguistic Inquiry* 46(1). 1–42.

Kubota, Yusuke & Robert Levine. 2014a. Scope anomaly of Gapping. In *Proceedings of NELS 44*, 247–260. GLSA.

Kubota, Yusuke & Robert Levine. 2014b. Unifying local and nonlocal modelling of respective and symmetrical predicates. In Glyn Morrill, Reinhard Muskens, Rainer Osswald & Frank Richter (eds.), *Proceedings of Formal Grammar 2014*, 104–120. Heidelberg: Springer.

Kubota, Yusuke & Robert Levine. 2015. Against ellipsis: arguments for the direct licensing of 'non-canonical' coordinations. *Linguistics and Philosophy* 38(6). 521–576.

Kubota, Yusuke & Robert Levine. 2016a. Gapping as hypothetical reasoning. *Natural Language and Linguistic Theory* 34(1). 107–156.

Kubota, Yusuke & Robert Levine. 2016b. The syntax-semantics interface of 'respective' predication: a unified analysis in Hybrid Type-Logical Categorial Grammar. *Natural Language and Linguistic Theory* 34(3). 911–973.

Kubota, Yusuke & Robert Levine. 2017. Pseudogapping as pseudo-VP ellipsis. *Linguistic Inquiry* 48(2). 213–257.

Kubota, Yusuke & Robert Levine. 2018. On an inadequate defense of an ellipsis-based analysis for right node raising. University of Tsukuba/Ohio State University ms.; available at https://ling.auf.net/lingbuzz/003852.

Levine, Robert. 2011. Linearization and its discontents. In Stefan Müller (ed.), *The proceedings of the 18th International Conference on Head-Driven Phrase Structure Grammar*, 126–146. Stanford: CSLI Publications. http://cslipublications.stanford.edu/%7BHPSG%7D/2011/.

Levine, Robert D. 2017. *Syntactic analysis: an HPSG-based approach*. Cambridge, England: Cambridge University Press.

Levine, Robert D. & Thomas E. Hukari. 2006. *The unity of unbounded dependency constructions* (CSLI Lecture Notes 166). Stanford, CA: CSLI Publications.

Lewis, Mike & Mark Steedman. 2013. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics* 1. 179–192.

Martin, Scott. 2013. *The dynamics of sense and implicature*. Ohio State University dissertation.

McCawley, James D. 1993. Gapping with shared operators. In David A. Peterson (ed.), *Berkeley Linguistics Society*, 245–253. Berkeley, California.

McCloskey, James. 1979. *Transformational syntax and model-theoretic semantics*. Dordrecht: Reidel.

Merchant, Jason. 2013. Ellipsis: a survey of analytical approaches. ms., University of Chicago.

Mihaliček, Vedrana & Carl Pollard. 2012. Distinguishing phenogrammar from tectogrammar simplifies the analysis of interrogatives. In Philippe de Groote & Mark-Jan Nederhof (eds.), *Formal Grammar 2010/2011*, 130–145. Heidelberg: Springer.

Miller, Philip. 2014. A corpus study of pseudogapping and its theoretical consequences. In Christopher Piñón (ed.), *Empirical issues in syntax and semantics*, vol. 10, 73–90.

Mineshima, Koji, Ribeka Tanaka, Pascual Martínez-Gómez, Yusuke Miyao & Daisuke Bekki. 2016. Building compositional semantics and higher-order inference system for a wide-coverage Japanese CCG parser. In *Proceedings of emnlp 2016*, 2236–2242. Austin, Texas: Association for Computational Linguistics.

Moortgat, Michael. 2011. Categorial type logics. In Johan F. A. K. van Benthem & G. B. Alice ter Meulen (eds.), *Handbook of logic and language*, 2nd edn., 95–179. Amsterdam: Elsevier.

Moortgat, Michael & Richard T. Oehrle. 1994. Adjacency, dependence, and order. In Paul Dekker & Martin Stokhof (eds.), *Proceedings of the Ninth Amsterdam Colloquium*, 447–466. Universiteit van Amsterdam: Instituut voor Taal, Logica, en Informatica.

Moot, Richard. 2015. A type-logical treebank for French. *Journal of Language Modelling* 3(1). 229–264.

Moot, Richard. 2017. Combining logical and distributional methods in type=logical grammars. *Journal of Language Modelling* 0(0). 00–00.

Moot, Richard & Christian Retoré. 2012. *The logic of categorial grammars: a deductive account of natural language syntax and semantics*. Heidelberg: Springer.

Morrill, Glyn. 2010. *Categorial grammar: logical syntax, semantics, and processing*. Oxford: Oxford University Press.

Morrill, Glyn. 2017. Grammar logicised: relativisation. *Linguistics and Philosophy* 40(2). 119–163.

Morrill, Glyn V. 1994. *Type Logical Grammars: Categorial logic of signs.* Dordrecht: Kluwer Academic Publishers.

Morrill, Glyn & Josep-Maria Merenciano. 1996. Generalizing discontinuity. *Traitement Automatique des Langues* 27(2). 119–143.

Morrill, Glyn & Teresa Solias. 1993. Tuples, discontinuity, and gapping in categorial grammar. In *Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics*, 287–296. Morristown, NJ, USA: Association for Computational Linguistics.

Morrill, Glyn & Oriol Valentín. 2017. A reply to Kubota and Levine on gapping. *Natural Language and Linguistic Theory* 35(1). 257–270.

Morrill, Glyn, Oriol Valentín & Mario Fadda. 2011. The displacement calculus. *Journal of Logic, Language and Information* 20(1). 1–48.

Muskens, Reinhard. 2001. Categorial Grammar and Lexical-Functional Grammar. In Miriam Butt & Tracy Holloway King (eds.), *The Proceedings of the LFG '01 Conference*. University of Hong Kong. http://csli-publications.stanford.edu/LFG/6/lfg01.html.

Muskens, Reinhard. 2003. Language, lambdas, and logic. In Geert-Jan Kruijff & Richard Oehrle (eds.), *Resource sensitivity in binding and anaphora*, 23–54. Dordrecht: Kluwer.

Newmeyer, Frederick J. 2016. Nonsyntactic explanations of island constraints. *Annual Review of Linguistics* 2. 187–210.

Oehrle, R. T. 1988. Multi-dimensional compositional functions as a basis for grammatical analysis. In R. T. Oehrle, E. Bach & D. Wheeler (eds.), *Categorial grammars and natural language structures*, 349–389. Dordrecht: Reidel.

Oehrle, Richard T. 1971. On Gapping. MS., MIT.

Oehrle, Richard T. 1987. Boolean properties in the analysis of gapping. In Geoffrey J. Huck & Almerindo E. Ojeda (eds.), *Syntax and semantics: discontinuous constituency*, vol. 20, 203–240. Academic Press.

Oehrle, Richard T. 1994. Term-labeled categorial type systems. *Linguistics and Philosophy* 17(6). 633–678.

Oehrle, Richard T. 2011. Multi-Modal Type-Logical Grammar. In Robert D. Borsley & Kersti Börjars (eds.), *Non-transformational syntax: Formal and explicit models of grammar: A guide to current models*, 225–267. Oxford, UK/Cambridge, MA: Blackwell Publishers Ltd.

Park, Sang-Hee, Jean-Pierre Koenig & Rui Chaves. 2018. Scope ambiguities and Lexical Resource Semantics: the case of Gapping. Paper presented at The 25th International Conference on Head-Driven Phrase Structure Grammar.

Partee, Barbara Hall. 1970. Negation, conjunction, and quantifiers: syntax vs. semantics. *Foundations of Language* 6. 153–165.

Partee, Barbara & Mats Rooth. 1983. Generalized conjunction and type ambiguity. In Rainer Bäuerle, Christoph Schwarze & Arnim von Stechow (eds.), *Meaning, use, and interpretation of language*, 361–383. Berlin: Walter de Gruyter.

Pickering, Martin & Guy Barry. 1991. Sentence processing without empty categories. *Language and Cognitive Processes* 6. 229–264.

Pickering, Martin & Guy Barry. 1993. Dependency Categorial Grammar and coordination. *Linguistics* 31(5). 855–902.

Pogodalla, Sylvain. 2017. A syntax-semantics interface for Tree-Adjoining Grammars through Abstract Categorial Grammars. *Journal of Language Modelling* 5(3). 527–605.

Pollard, Carl. 2013. Linear categorial grammar. ms., Lecture Notes at ESSLLI 2013.

Pollard, Carl. 2014. What numerical determiners mean: a non-ambiguity analysis. Talk presented at the Workshop on Semantics of Cardinals, Ohio State University, March 6, 2014.

Pollard, Carl J. & Ivan A. Sag. 1987. *Information-based syntax and semantics* (CSLI Lecture Notes 13). Stanford, CA: CSLI Publications.

Puthawala, Daniel. 2018. Stripping isn't so mysterious, or anomalous scope, either. In Annie Foret, Greg Kobele & Sylvain Pogodalla (eds.), *Formal Grammar 2018*, 102–120.

Restall, Greg. 2000. *An introduction to substructural logics*. London: Routledge.

Sag, Ivan A. 2010. English filler-gap constructions. *Language* 86(3). 486–545.

Sag, Ivan A., Hans C. Boas & Paul Kay. 2012. Introducing Sign-Based Construction Grammar. In Hans C. Boas & Ivan A. Sag (eds.), *Sign-Based Construction Grammar* (CSLI Lecture Notes 193), 1–29. Stanford, CA: CSLI Publications.

Sag, Ivan A., Gerald Gazdar, Thomas Wasow & Steven Weisler. 1985. Coordination and how to distinguish categories. *Natural Language and Linguistic Theory* 3(2). 117–171.

Siegel, Muffy A. 1987. Compositionality, case, and the scope of auxiliaries. *Linguistics and Philosophy* 10(1). 53–75.

Steedman, Mark. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory* 5(3). 403–439.

Steedman, Mark. 2012. *Taking scope*. Cambridge, Mass.: MIT Press.

Steedman, Mark J. 1985. Dependency and coordination in the grammar of Dutch and English. *Language* 61(3). 523–568.

Steedman, Mark J. 1990. Gapping as constituent coordination. *Linguistics and Philosophy* 13(2). 207–265.

Steedman, Mark J. 1997. *Surface structure and interpretation* (Linguistic Inquiry Monographs 30). Cambridge, MA: MIT Press.

Steedman, Mark J. 2000. *The syntactic process* (Language, Speech, and Communication). Cambridge, MA: MIT Press.

Steedman, Mark & Jason Baldridge. 2011. Combinatory Categorial Grammar. In Robert D. Borsley & Kersti Börjars (eds.), *Non-transformational syntax: Formal and explicit models of grammar: A guide to current models*, 181–224. Oxford, UK/Cambridge, MA: Blackwell Publishers Ltd.

Szabolcsi, Anna. 1992. Combinatory grammar and projection from the lexicon. In Ivan A. Sag & Anna Szabolcsi (eds.), *Lexical matters* (CSLI Lecture Notes 24), 241–268. Stanford: CSLI Publications.

Vermaat, Willemijn. 2005. *The logic of variation. a cross-linguistic account of wh-question formation*. Utrecht University dissertation.

Worth, Christopher. 2016. *English coordination in Linear Categorial Grammar*. Ohio State University dissertation.

Yatabe, Shûichi. 2001. The syntax and semantics of Left-Node Raising in Japanese. In Daniel P. Flickinger & Andreas Kathol (eds.), *Proceedings of the HPSG-2000 Conference, University of California, Berkeley*, 325–344. CSLI Publications. http://csli-publications.stanford.edu/HPSG/1/, accessed 2018-2-25.

Yatabe, Shûichi & Wai Lok Tam. 2017. In defense of an HPSG-based theory of non-constituent coordination: a reply to Kubota and Levine. ms., available at http://ling.auf.net/lingbuzz/003152.

Yoshida, Masaya, Tim Hunter & Michael Frazier. 2015. Parasitic gaps licensed by elided syntactic structure. *Natural Language and Linguistic Theory* 33(4). 1439–1471.

Zaenen, Annie. 1983. On syntactic binding. *Linguistic Inquiry* 14(3). 469–504.