Coordination, ellipsis and Type-Logical Categorial Grammar (Working Title)

Yusuke Kubota
University of Tsukuba
Ohio State University
kubota.yusuke.fn@u.tsukuba.ac.jp

Robert Levine
Ohio State University
levine@ling.ohio-state.edu

May 24, 2015

Preface

This is a chapter draft of a book in preparation that presents a new theory of natural language syntax called Hybrid Type-Logical Categorial Grammar (Hybrid TLCG). Hybrid TLCG was first proposed in Yusuke Kubota's dissertation (Kubota 2010; [1] in the list below), and has since been developed by the present two authors. As of writing (May 2015), there are five journal articles [2,3,6,7,8], one manuscript under review for publication [9], and two conference papers [4,5], which use Hybrid TLCG as the underlying framework for analyzing complex empirical phenomena such as Gapping, pseudogapping, nonconstituent coordination, complex predicates and the semantics of symmetrical predicates (*same*/*different*).

Though we have made every effort to present our framework clearly in each of these papers, space limitations have made it difficult to explain all the important details in journal article and conference paper formats. Categorial grammar, after all, involves somewhat forbidding technical contents that are not necessarily familiar to linguists, and Hybrid TLCG in particular belongs to a tradition which takes the logical underpinnings of the theory very seriously.

The present document unpacks the relevant details in a (hopefully) considerably more leisurely and accessible format than in previously published material of ours, aiming to provide a gentle introduction to Hybrid TLCG to syntacticians and semanticists who do not have any background in categorial grammar. The presentation is specifically pitched to readers who are trained in and familiar with the standard derivational approach to syntax. (We would also like to mention that many notions we explain along the way are not specific to Hybrid TLCG, and hence, we believe that this document will serve as a general introduction to categorial grammar as well.)

This is the second version of the draft (lightly revised from the first version from November 2014), and though we believe that it is already useful for readers who have found the above articles challenging to read, we anticipate that there are still several places where further improvements can be made. So, comments on any aspect of this document are earnestly solicited and would be very much appreciated. Please send us feedback (emails: kubota.yusuke.fn@u.tsukuba.ac.jp, levine.1@osu.edu), and we'll further revise and improve the presentation.

Related works:

[1] Kubota, Yusuke. 2010. (In)flexibility of Constituency in Japanese in Multi-Modal Categorial Grammar with Structured Phonology. Ph.D. thesis, Ohio State University. http://www.u.tsukuba.ac.jp/~kubota.yusuke.fn/papers/kubota-diss.pdf

[2] Kubota, Yusuke. 2014. The logic of complex predicates: A deductive synthesis of 'argument sharing' and 'verb raising'. *NLLT* 32(4) 1145-1204. http://link.springer.com/article/10.1007/s11049-014-9246-8

[3] Kubota, Yusuke. 2015. Nonconstituent coordination in Japanese as constituent coordination: An analysis in Hybrid Type-Logical Categorial Grammar. *Linguistic Inquiry* 46(1) 1–42. http://www.mitpressjournals.org/doi/abs/10.1162/LING_a_00174#.VWJxvk9_NBc

[4] Kubota, Yusuke and Robert Levine. 2014. Pseudogapping as pseudo-VP ellipsis. In *LACL 2014*. http://www.u.tsukuba.ac.jp/~kubota.yusuke.fn/papers/kl-fg2014.pdf

[5] Kubota, Yusuke and Robert Levine. 2014. Unifying local and nonlocal modelling of respective and symmetrical predicates. In *Formal Grammar 2014*. http://www.u.tsukuba.ac.jp/~kubota.yusuke.fn/papers/kl-pseudo.pdf

[6] Kubota, Yusuke and Robert Levine. 2014. Gapping as hypothetical reasoning. To appear in *NLLT*. http://ling.auf.net/lingbuzz/002123

[7] Kubota, Yusuke and Robert Levine. 2015. Against ellipsis: Arguments for the direct licensing of 'non-canonical' coordinations. To appear in *Linguistics and Philosophy*. http://ling.auf.net/lingbuzz/002214

[8] Kubota, Yusuke and Robert Levine. 2015. A unified analysis of 'respective', symmetrical and summative predicates. To appear in *NLLT*. http://ling.auf.net/lingbuzz/002150

[9] Kubota, Yusuke and Robert Levine. 2015. Pseudogapping as pseudo-VP ellipsis. MS. http://ling.auf.net/lingbuzz/002504

Chapter 3

Hybrid Type-Logical Categorial Grammar

[DRAFT; comments welcome]

3.1 Introduction

In this chapter, we introduce Hybrid Type-Logical Categorial Grammar (Hybrid TLCG; Kubota 2010, 2014, 2015; Kubota and Levine 2012, 2013a,b, 2014b,a). Hybrid TLCG is a variant of categorial grammar (CG) that belongs to the tradition of Type-Logical Categorial Grammar (Morrill 1994; Moortgat 1997). We provide a gentle introduction of our framework for syntacticians and semanticists without prior familiarity with CG. Although familiarity with notions in (discrete) mathematics, logic and computer science would be helpful, we will try to make the ensuing exposition accessible to anybody with background in graduate introductory level syntax and (formal) semantics.

Since we expect the majority of our readers to be familiar with the currently standard 'derivational' (or 'movement-based') model of syntax, the exposition below is tailored to this audience (thus, logically-savvy readers may be a bit puzzled and disoriented by the order of presentation). We start with a simple variant of CG called the AB grammar, which is essentially equivalent to unadorned phrase structure grammar—that is, simple context free grammar without any movement operations or devices that mimic movement such as the SLASH feature in G/HPSG. The AB grammar we present below has a number of important characteristics that are common to most variants of CG, including the distinction between forward and backward slashes to model word order, and the homomorphic mapping from syntactic categories to semantic types, the latter of which is at the heart of the simple and systematic syntax-semantics interface of CG.

After explaining these basic properties of CG with the AB grammar, we extend it with a new type of slash that is insensitive to word order (unlike the forward and backward slashes). This new slash is called the vertical slash and is a central component of Hybrid TLCG; by just adding this single new mechanism, we can model both overt movement and covert movement from the standard derivational approach within CG. We illustrate the workings of this extended system with the basic analyses of quantification (for 'covert movement')

and topicalization (for 'overt movement'). The analyses crucially exploit the notion of HYPOTHETICAL REASONING that is characteristic of the TYPE-LOGICAL variants of CG (as opposed to 'rule-based' systems such as Combinatory Categorial Grammar (CCG; Steedman 1996, 2000); see Sect. 3.8.1 for a comparison with CCG).

In the final part of the framework exposition, we extend the system further by introducing a mechanism, originally due to Lambek (1958), that allows for hypothetical reasoning for the directional (i.e. forward and backward) slashes as well. Hypothetical reasoning for directional slashes enables reanalyzing substrings of sentences as constituents, and supports elegant analyses of various 'nonconstituent' phenomena including (but not limited to) nonconstituent coordination. In this chapter, we illustrate this feature of our framework with the analyses of Right-Node Raising and Dependent Cluster Coordination, two major types of nonconstituent coordination in English. We show in particular how the simple analysis of the syntax of these phenomena in CG supports a straightforward account of the interaction between nonconstituent coordination and the scope of generalized quantifiers. At the heart of this analysis is the 'hybrid' implication system of the present framework, where inferences involving directional slashes freely interact with inferences involving the non-directional, vertical slash. Later chapters exploit this property of Hybrid TLCG more extensively, in the analyses of more complex linguistic phenomena, especially Gapping and the so-called 'respective' readings and related phenomena (including the semantics of *same* and *different*). Later chapters will also illustrate how the flexible notion of constituency is useful not only in the analyses of standard types of nonconstituent coordination (as is already well-known in the CG literature) but also in characterizing elliptical linguistic expressions in Gapping and pseudogapping.

The empirical phenomena we treat in this book have all posed recalcitrant problems for both derivational and nonderivational approaches in the previous syntactic literature. We show how the flexible and systematic syntax-semantics interface of Hybrid TLCG enables simple analyses of these phenomena that overcome the problems of previous proposals while at the same time integrating the key analytic insights from these previous proposals.

3.2 The AB grammar

We start with a simple fragment of CG called the AB GRAMMAR, and introduce some key concepts and notations along the way that are shared by many variants of CG. Following the standard practice in CG, linguistic expressions are written as triples of prosodic representation (i.e., 'PF'—we gloss over fine details of this component, so this is just string of words for all practical purposes), semantic interpretation and syntactic category, notated $\langle \pi; \sigma; \gamma \rangle$ (angle brackets will be omitted below). We assume that the set of syntactic categories is infinite and is recursively defined. This assumption reflects the view that natural language syntax is a kind of logic, a perspective that is particularly salient in the 'type-logical' variants of CG (which Hybrid TLCG is an instance of). The underlying idea is that syntactic categories are like formulas in propositional logic. Just as the set of well-formed formulas in propositional logic is infinite in order to deal with unboundedly complex logical inferences, syntactic categories are infinite in order to deal with (potentially) unboundedly complex combinatory properties of linguistic expressions.

We start with the following definition of SYNTACTIC CATEGORIES (or SYNTACTIC TYPES;

following the convention in TLCG, we use the terms 'syntactic type' and 'syntactic category' interchangeably):[1]

(1) a. N, NP and S are categories.
b. If A and B are categories, then so are A/B and $B\backslash A$.
c. Nothing else is a category.

The categories in (1a) are called ATOMIC CATEGORIES. The ones involving slashes built from atomic categories via clause (1b) are called COMPLEX CATEGORIES. Note that this definition allows for an infinite set of syntactic categories. For example, by (1b) with $A = B =$ NP, we have NP/NP as a category. Applying (1b) again, this time with $A = B =$ NP/NP, yields (NP/NP)/(NP/NP). Doing the same thing once again yields ((NP/NP)/(NP/NP))/((NP/NP)/(NP/NP)). And so on. The definition of syntactic categories becomes slightly more complex once we introduce the vertical slash ($\uparrow$) below (see Sect. 3.10.1 for a complete definition), but for now, the simple definition in (1) (standard in the CG literature) suffices.

One important feature of CG is that it lexicalizes the valence (or subcategorization) properties of linguistic expressions via the use of complex syntactic categories. 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):

(2) a. ran; NP\S
b. read; (NP\S)/NP

(2a) says that the verb *ran* combines with its argument NP *to its left* to become an S. Likewise, (2b) says that *read* first combines with an NP *to its right* and then another NP to its left to become an S. Thus, the slashes encode both the linear order in which the verb looks for its arguments and the number (and type) of arguments that they subcategorize for. It is useful to introduce some terminology here: we say that complex categories (like the ones in (2)) designate FUNCTORS that combine with ARGUMENTS to return RESULTS.

As already noted, the distinction between the FORWARD SLASH (/) and the BACKWARD SLASH (\) corresponds to the distinction in the directions (in the surface word order) in which functors look for arguments. That is, A/B is a functor looking for a B *to its right* (to become an A) whereas $B\backslash A$ is a functor looking for a B *to its left*. We adopt the so-called Lambek-style notation for syntactic categories, in which arguments are always written 'under the slash' (in the alternative notation adopted in CCG, the backward slash is written in the opposite way, where our $B\backslash A$ will be written $A\backslash B$). We omit outermost parentheses and parentheses for a sequence of the same type of slash, assuming that / and $\uparrow$ (introduced below) are left associative, and \ right associative. Thus, S/NP/NP, NP\NP\S and S$\uparrow$NP$\uparrow$NP are abbreviations of ((S/NP)/NP), (NP\(NP\S)) and ((S$\uparrow$NP)$\uparrow$NP), respectively.

We first introduce the two basic rules in the grammar, namely, SLASH ELIMINATION rules for forward and backward slashes. In the so-called 'labelled deduction' format of natural deduction (cf. Oehrle 1994; Morrill 1994), these rules are formulated as in (3).

[1] We assume a small number of syntactic features for atomic categories, notated as subscripts as in NP_n (for nominative NP) and NP_a (for accusative NP).

(3) a. FORWARD SLASH ELIMINATION

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

b. BACKWARD SLASH ELIMINATION

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

The inputs to the rule are written above the horizontal line and the output is written below the line. In line with the analogy between language and logic underlying TLCG, we call the inputs of these rules PREMISES and the outputs CONCLUSIONS. The linear order between the two premises above the line has only mnemonic significance (as will become clear below, what *is* significant to word order is instead the order of a and b in the prosody of the expression obtained as the conclusion). The labelled deduction presentation is so called since, in addition to the syntactic categories of the premises and conclusions, the rules are also annotated (or labelled) with how the semantics and prosody of the conclusion are computed given the semantics and prosody of the premises.

The PROOF (or DERIVATION; following the CG tradition, we use these two terms interchangeably but it should be kept in mind that the notion of derivation in CG is quite different from that in standard derivational approaches) in (4) illustrates how larger linguistic expressions are built from smaller ones using the rules just introduced. Here, a transitive verb, of category (NP\S)/NP, is combined with its two arguments on the right (object) and left (subject).

(4)

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

Note that the object NP *Mary* is placed to left of the verb in the proof tree, but this does not have any significance in the linear order of the string derived. Thus, unlike linguistic trees in ordinary syntactic theories (or in CCG), the left-to-right yield of the proof trees does not correspond to word order.

From a 'logical' point of view, the two slashes should 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 should be thought of as directional variants of MODUS PONENS ($B \to A, B \vdash A$). Thus, (4) is literally a proof of the fact that a complete sentence exists if there is one NP to the right and one NP to the left of the verb *loves*.

We now turn to the syntax-semantics mapping. For expository ease, we assume a standard Montagovian model-theoretic semantics.[2] As is standard in CG, we assume a homomorphic mapping from syntactic categories to semantic types.[3] This means that, for each linguistic expression, given its syntactic category, one can determine uniquely and unambiguously its semantic type (e.g. individuals, sets of individuals, sets of sets of

[2]Since we do not deal with phenomena that crucially involve intensionality, we assume an extensionalized fragment throughout. Also, our approach is in principle compatible with more sophisticated semantic theories such as dynamic semantics. See, for example, Martin (2013) and Martin and Pollard (2014) for a proposal incorporating a compositional dynamic semantics to Linear Categorial Grammar (LCG), a version of CG closely related to Hybrid TLCG.

[3]LINEAR CATEGORIAL GRAMMAR (Mihaliček and Pollard 2012; Worth 2014) is different from other variants of CG in that it does not assume a functional mapping from syntactic types to semantic types. Instead the mapping is relational (see in particular Worth (2014, section 1.1) for an explicit statement of this point).

individuals, etc.). To work this out, we first assume that semantic types are recursively built from the basic types e (individuals) and t (truth values) in the standard fashion:[4]

(5) a. e and t are semantic types.
b. If α and β are semantic types, then so is $\alpha \to \beta$.
c. Nothing else is a semantic type.

Then, we can define the function Sem that returns, for each syntactic category given as input, its semantic type:

(6) (Base Case)
a. $\mathsf{Sem}(\mathrm{NP}) = e$
b. $\mathsf{Sem}(\mathrm{N}) = e \to t$
c. $\mathsf{Sem}(\mathrm{S}) = t$

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

(7) says that the semantic type of a complex syntactic category A/B (or $B\backslash A$) is that of a function that takes as its argument an expression that has the semantic type of its syntactic argument (i.e. $\mathsf{Sem}(B)$) and returns an expression that has the semantic type of its result syntactic category (i.e. $\mathsf{Sem}(A)$).

We can now write full lexical entries that specify the semantics as well:

(8) a. ran; **ran**; NP\S
b. saw; **saw**; (NP\S)/NP

Given the functional mapping from syntactic categories to semantic types in (7), we see that the intransitive verb *ran* is semantically of type $e \to t$ (set of individuals) and that the transitive verb *saw* is of type $e \to (e \to t)$ (a function from individuals to sets of individuals, i.e., a curried two-place relation).

Syntactic rules with semantics can then be written as in (9) and a sample derivation with semantic annotation is given in (10).

(9) a. Forward Slash Elimination

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

b. Backward Slash Elimination

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

(10)

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

[4] We replace Montague's (1973) notation $\langle \beta, \alpha \rangle$ with $\beta \to \alpha$, so that the notation more transparently reflects the fact that the complex type is a functional type.

Note that the semantic effect of Slash Elimination is FUNCTION APPLICATION in both of the two rules in (9).

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).

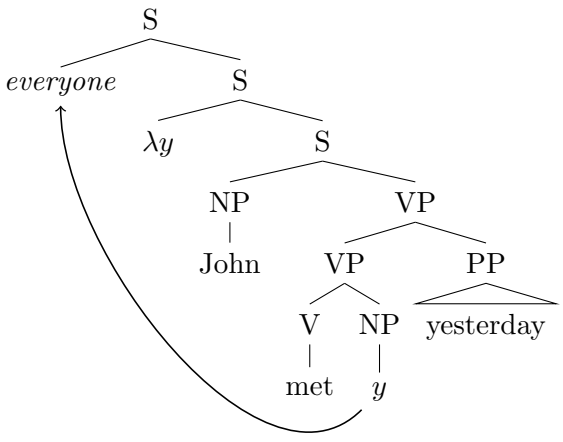
3.3 Adding the vertical slash to the AB grammar

The AB grammar introduced above is basically equivalent to phrase structure grammar without elaborate additional mechanisms such as (overt or covert) movement, the SLASH feature inheritance mechanism in G/HPSG, or quantifier storage due to Cooper (1983). In fact, the formal equivalence between the AB grammar and context free grammar (i.e. simple PSG) was proved as early as Bar-Hillel et al. (1960). This means that the AB grammar is too impoverished for modelling complex empirical phenomena in natural language that have motivated the notion of 'movement' in transformational grammar and various alternative mechanisms (like the SLASH feature and Cooper storage) in other frameworks.

In this section, we extend the AB grammar by adding just one new mechanism—specifically, a new type of slash called the VERTICAL SLASH ($\uparrow$), which, unlike the forward and backward slashes, does not encode linear order in syntactic categories. This extended fragment can naturally model the notion of movement in derivational approaches. The mechanism of vertical slash in Hybrid TLCG is moreover more general than that of movement in derivation approaches, and this difference becomes crucial in the analysis of Gapping we present in chapter 4. We will come back to this point in that chapter.

The problem with the AB grammar can be illustrated by the failed derivation for the sentence *John met everyone yesterday* in (12), which attempts to mimic the QR-based analysis in a derivational framework in (11).

(11)



(12)

$$\dfrac{\text{everyone; NP} \quad \dfrac{\text{john; NP} \quad \dfrac{\dfrac{\text{met; (NP\S)/NP} \quad \varphi\text{; NP}}{\text{met} \circ \varphi\text{; NP\S}}\,/\text{E} \quad \dfrac{\text{yesterday;}}{\text{(NP\S)\(NP\S)}}}{\text{met} \circ \varphi \circ \text{yesterday; NP\S}}\,\backslash\text{E}}{\text{john} \circ \text{met} \circ \varphi \circ \text{yesterday; S}}\,\backslash\text{E}}{\text{john} \circ \text{met} \circ \text{everyone} \circ \text{yesterday; S}}\,????$$

In (12), to model the covert movement of the quantifier, we start by positing a dummy expression, whose prosody is represented by the symbol φ. The derivation proceeds up to the point where a complete sentence is formed containing this 'trace'. What we then need to do at this point is to combine the quantifier with this sentence so that it semantically scopes over the whole sentence but appears in the 'trace' position in the prosodic form (since this is an instance of 'covert movement').

However, things go wrong in all of the three components of the tripartite linguistic sign. First, syntactically, with the two Slash Elimination rules we have introduced above, we can only combine functors (i.e. expressions of the form A/B or $B\backslash A$, involving either the forward or the backward slash) with arguments, but neither S nor NP is a functor that takes the other as its argument. Prosodically, all we can do with the two rules we have introduced above is to concatenate the two strings. But here, what we need to do is a more elaborate type of string manipulation whereby we replace the dummy string φ with the string of the quantifier *everyone* (i.e. an operation corresponding to Montague's (1973) syncategorematic rule of 'quantifying-in'). Finally, semantically, the variable corresponding to the 'trace' needs to be bound by the λ-operator and the property thus obtained be given as an argument to the quantifier, but we currently do not have any rule in the grammar that allows for such a complex meaning assembly.

Are we then stuck? In a sense, yes, since the above illustration clearly shows that there is no natural way of modelling the notion of covert movement in the AB grammar. Essentially the same point can be made with overt movement. But in fact, our attempt above with the failed derivation in (12) was *almost* on the right track. Specifically, all that is missing is a mechanism that enables the last step to go through. And it turns out that such a mechanism is already available in CG, albeit outside of the AB grammar and in a relatively recent line of development which has not yet gained due recognition in the linguistic literature.

Although approaches that distinguish word order by the forward and backward slashes (like the AB grammar fragment formulated above) have been the mainstream in CG research, there is a relatively recent strand of research that relocates the word order information from the syntactic types to the prosodic component (Oehrle 1994; de Groote 2001; Muskens 2003; Mihaliček and Pollard 2012). Following Pollard (2013), we call this family of approaches Linear Categorial Grammar (LCG). LCG is characteristic for its use of the λ-calculus for modelling the prosody of linguistic expressions, and this plays a key role in encoding word order information in the prosodic representation directly.

Part of the motivation for LCG comes precisely from the recognition that variants of CG based on the distinction between forward and backward slashes are suboptimal for handling phenomena that are essentially insensitive to word order, such as quantification and extraction (see Muskens (2003) for a particularly lucid discussion on this point). We incorporate the key insight of LCG into our AB fragment, and show that the extended fragment indeed gives us exactly the right tool to 'fill in the gap' in the above derivation in (12). The reason we make this move (of combining AB with LCG), rather than simply switching to LCG will become clear later—it mainly has to do with the analysis of coordination. In a purely 'non-directional' calculus of LCG, the analysis of coordination quickly becomes extremely unwieldy, and, so far as we are aware, this issue has not been addressed in any detail in the LCG literature. See Sect. 3.8.2 for a more detailed discussion on this point.

The new mechanism we incorporate from LCG is an order-insensitive mode of implication $\uparrow$ called VERTICAL SLASH. We introduce the INTRODUCTION and ELIMINATION rules for this slash, which are formulated as follows (as with /, we write the argument to the right for $\uparrow$; the harpoon is there as a visual aide indicating that the right category is the argument):

(13) a. VERTICAL SLASH INTRODUCTION

$$\dfrac{\begin{array}{ccccccc}\vdots & \vdots & [\varphi;\, x;\, A]^n & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ & & b;\, \boldsymbol{\mathcal{F}};\, B & & \end{array}}{\lambda\varphi.b;\, \lambda x.\boldsymbol{\mathcal{F}};\, B{\uparrow}A}\ {\uparrow}\mathrm{I}^n$$

b. VERTICAL SLASH ELIMINATION

$$\dfrac{a;\, \boldsymbol{\mathcal{F}};\, A{\uparrow}B \qquad b;\, \boldsymbol{\mathcal{G}};\, B}{a(b);\, \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\, A}\ {\uparrow}\mathrm{E}$$

The workings of these rules can be best illustrated with examples. We show in (14) the derivation for the sentence *John saw everyone yesterday.*

(14)

$$\textcircled{3}\rightarrow \dfrac{\dfrac{\lambda\sigma.\sigma(\mathsf{everyone});\ \mathbf{V}_{\mathbf{person}};\ \mathrm{S}{\uparrow}(\mathrm{S}{\uparrow}\mathrm{NP}) \qquad \textcircled{2}\rightarrow \dfrac{\textcircled{1}\rightarrow \dfrac{\mathsf{john};\ \mathbf{j};\ \mathrm{NP} \qquad \dfrac{\dfrac{\mathsf{saw};\ \mathbf{see};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP} \qquad \left[\begin{array}{l}\varphi;\\ x;\ \mathrm{NP}\end{array}\right]^1}{\mathsf{saw}\circ\varphi;\ \mathbf{see}(x);\ \mathrm{NP}\backslash\mathrm{S}}\ /\mathrm{E} \qquad \dfrac{\mathsf{yesterday};\ \mathbf{yest};\ (\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S})}{\mathsf{saw}\circ\varphi\circ\mathsf{yesterday};\ \mathbf{yest}(\mathbf{see}(x));\ \mathrm{NP}\backslash\mathrm{S}}\ \backslash\mathrm{E}}{\mathsf{john}\circ\mathsf{saw}\circ\varphi\circ\mathsf{yesterday};\ \mathbf{yest}(\mathbf{see}(x))(\mathbf{j});\ \mathrm{S}}\ \backslash\mathrm{E}}{\lambda\varphi.\mathsf{john}\circ\mathsf{saw}\circ\varphi\circ\mathsf{yesterday};\ \lambda x.\mathbf{yest}(\mathbf{see}(x))(\mathbf{j});\ \mathrm{S}{\uparrow}\mathrm{NP}}\ {\uparrow}\mathrm{I}^1}{\lambda\sigma.[\sigma(\mathsf{everyone})](\lambda\varphi.\mathsf{john}\circ\mathsf{saw}\circ\varphi\circ\mathsf{yesterday});\ \mathbf{V}_{\mathbf{person}}(\lambda x.\mathbf{yest}(\mathbf{see}(x))(\mathbf{j}));\ \mathrm{S}}\ {\uparrow}\mathrm{E}$$

$$\lambda\varphi.[\mathsf{john}\circ\mathsf{saw}\circ\varphi\circ\mathsf{yesterday}](\mathsf{everyone});\ \mathbf{V}_{\mathbf{person}}(\lambda x.\mathbf{yest}(\mathbf{see}(x))(\mathbf{j}));\ \mathrm{S}$$

$$\mathsf{john}\circ\mathsf{saw}\circ\mathsf{everyone}\circ\mathsf{yesterday};\ \mathbf{V}_{\mathbf{person}}(\lambda x.\mathbf{yest}(\mathbf{see}(x))(\mathbf{j}));\ \mathrm{S}$$

The main new ingredient here is a type of inference called HYPOTHETICAL REASONING. In ordinary kinds of logic, hypothetical reasoning is a type of proof in which one draws the conclusion $A \rightarrow B$ on the basis of a proof of B by first hypothetically assuming A. As will become clear below (especially when we introduce hypothetical reasoning for the directional slashes in Sect. 3.4), hypothetical reasoning is a very powerful (yet systematic) tool that is deeply rooted in the CG conception of natural language syntax as a kind of logic and which distinguishes CG from phrase structure-based theories of syntax (including both derivational and nonderivational approaches).

In (14), we first hypothetically assume an object NP with prosody φ and semantics x. In derivations, hypotheses are indicated by square brackets and are indexed. The indices are for keeping track of where the hypothesis is withdrawn in the whole proof—see below for more on this. The derivation proceeds in the same way as in (12) above up to the point where the whole sentence is formed (①). At this point, we know that the string containing the prosodic variable φ is a full-fledged sentence. We then apply the VERTICAL SLASH INTRODUCTION rule (13a) to withdraw this hypothesis (②). In this step, we are essentially concluding that the string *John saw* __ *yesterday* is a well-formed sentence of English *if* there is an NP in the gap position __. Note that this conclusion itself does not depend on the assumption that there is an NP in the object position. It is in this sense that Vertical Slash Introduction *withdraws* the hypothesis posited at an earlier step in the proof. The hypothesis is there only for the sake of drawing this conclusion and it doesn't play any other role in the grammar (and in this sense, it has a very different status

from traces in derivational approaches, which are representational objects). The $\upharpoonright$I step is coindexed with the hypothesis that is withdrawn so as to keep track of which hypothesis is withdrawn at which step in the proof. The vertical dots around the hypothesis in the rule in (13a) abbreviate an arbitrarily complex proof structure. Thus, (13a) simply says that a hypothesis posited at some previous step can be withdrawn by $\upharpoonright$I at any step in the proof.

Let us now examine the effect of the Vertical Slash Introduction rule more closely. In the prosody of the derived expression, the gap position is explicitly represented by the prosodic variable φ bound by the λ-operator. Correspondingly, the semantic action of $\upharpoonright$I is also variable binding—the variable x posited as the semantic placeholder for the 'trace' is bound. Note that this creates the right property to be given as an argument to the quantifier. Finally, the syntactic category S$\upharpoonright$NP, with the vertical slash $\upharpoonright$, indicates that the whole expression is a sentence missing an NP ('if there is an NP, then there is an S'), just like S/NP and NP\S. The difference from the directional slashes is that the vertical slash does not indicate where within the whole string the NP is missing, since that information is represented in the prosodic term itself via the use of λ-binding.

The quantifier then takes this function from strings into strings (of type **st** → **st**; with **st** the type of strings) as an argument and embeds its string in the gap position (③). This step is licensed by the VERTICAL SLASH ELIMINATION rule (13b). The dotted lines show reduction steps for the prosodic term (we often omit these in the derivations below) and should not be confused with the application of logical rules (of Slash Elimination and Introduction) designated by solid lines; unlike the latter, purely from a formal perspective, these reduction steps are redundant. Since the quantifier takes an **st** → **st** function as its argument, it has a higher-order prosody (of type (**st** → **st**) → **st**) itself. Semantically, the quantifier denotes a standard GQ meaning of type $(e \to t) \to t$. $\mathbf{V}_{\mathbf{person}}$ abbreviates the term $\lambda P.\forall x[\mathbf{person}(x) \to P(x)]$ (similarly for the existential quantifier $\mathbf{\exists}_{\mathbf{person}}$). Since the $\upharpoonright$E rule does function application both in the semantic and prosodic components, the right string/meaning pair is assigned to the whole sentence by this derivation.

The analysis of scope ambiguity is then straightforward, and is parallel to quantifying-in and QR. (15) shows the derivation for the inverse scope ($\forall > \exists$) reading of *Someone talked to everyone yesterday*:

(15)

$$\dfrac{\mathsf{talked} \circ \mathsf{to};\ \mathbf{talked\text{-}to};\ (\mathrm{NP\backslash S})/\mathrm{NP} \qquad [\varphi_1; x_1; \mathrm{NP}]^1}{\mathsf{talked} \circ \mathsf{to} \circ \varphi_1;\ \mathbf{talked\text{-}to}(x_1);\ \mathrm{NP\backslash S}}\ /\mathrm{E}$$

$$\dfrac{[\varphi_2; x_2; \mathrm{NP}]^2 \qquad \mathsf{talked} \circ \mathsf{to} \circ \varphi_1;\ \mathbf{talked\text{-}to}(x_1);\ \mathrm{NP\backslash S}}{\varphi_2 \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1;\ \mathbf{talked\text{-}to}(x_1)(x_2);\ \mathrm{S}}\ \backslash\mathrm{E}$$

$$\dfrac{\varphi_2 \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1;\ \mathbf{talked\text{-}to}(x_1)(x_2);\ \mathrm{S} \qquad \mathsf{yesterday};\ \mathbf{yest};\ \mathrm{S\backslash S}}{\varphi_2 \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2));\ \mathrm{S}}\ \backslash\mathrm{E}$$

$$\dfrac{\varphi_2 \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2));\ \mathrm{S}}{\lambda\varphi_2.\varphi_2 \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2));\ \mathrm{S}\upharpoonright\mathrm{NP}}\ \upharpoonright\mathrm{I}^2$$

$$\dfrac{\lambda\sigma.\sigma(\mathsf{someone});\ \mathbf{\exists}_{\mathbf{person}};\ \mathrm{S}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP}) \qquad \lambda\varphi_2.\varphi_2 \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2));\ \mathrm{S}\upharpoonright\mathrm{NP}}{\mathsf{someone} \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)));\ \mathrm{S}}\ \upharpoonright\mathrm{E}$$

$$\dfrac{\mathsf{someone} \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)));\ \mathrm{S}}{\lambda\varphi_1.\mathsf{someone} \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \lambda x_1.\mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)));\ \mathrm{S}\upharpoonright\mathrm{NP}}\ \upharpoonright\mathrm{I}^1$$

$$\dfrac{\lambda\sigma.\sigma(\mathsf{everyone});\ \mathbf{V}_{\mathbf{person}};\ \mathrm{S}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP}) \qquad \lambda\varphi_1.\mathsf{someone} \circ \mathsf{talked} \circ \mathsf{to} \circ \varphi_1 \circ \mathsf{yesterday};\ \lambda x_1.\mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)));\ \mathrm{S}\upharpoonright\mathrm{NP}}{\mathsf{someone} \circ \mathsf{talked} \circ \mathsf{to} \circ \mathsf{everyone} \circ \mathsf{yesterday};\ \mathbf{V}_{\mathbf{person}}(\lambda x_1.\mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2))));\ \mathrm{S}}\ \upharpoonright\mathrm{E}$$

The point here is that the scopal relation between multiple quantifiers depends on the order of application of the hypothetical reasoning involving $\uparrow$ to introduce quantifiers. We get the inverse scope reading in this derivation since the subject quantifier is introduced in the derivation first.

In the present approach, the difference between overt movement and covert movement comes down to a lexical difference in the 'prosodic action' of the operator that triggers the 'movement' operation in question. As shown above, covert movement is modelled by an operator which embeds some (non-empty) string in the gap position, whereas overt movement is modelled by an operator which embeds the empty string in the gap position. Thus, as shown by Muskens (2003), extraction can be analyzed quite elegantly in this type of CG with hypothetical reasoning for $\uparrow$, as in the derivation in (17) for the topicalization example (16).

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

(17)

$$\dfrac{\mathsf{bagels};\ \mathbf{b};\ \mathrm{NP} \qquad \dfrac{\lambda\sigma\lambda\varphi.\varphi\circ\sigma(\boldsymbol{\epsilon});\ \lambda\mathcal{F}.\mathcal{F};\ (\mathrm{S}{\uparrow}X){\uparrow}(\mathrm{S}{\uparrow}X) \qquad \textcircled{1}\rightarrow\ \dfrac{\mathsf{kim};\ \mathbf{k};\ \mathrm{NP} \qquad \dfrac{\dfrac{\mathsf{gave};\ \mathbf{gave};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{PP}/\mathrm{NP} \qquad \left[\begin{array}{l}\varphi;\\ x;\\ \mathrm{NP}\end{array}\right]^1}{\mathsf{gave}\circ\varphi;\ \mathbf{gave}(x);\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{PP}}\,/\mathrm{E} \qquad \begin{array}{l}\mathsf{to}\circ\mathsf{chris};\\ \mathbf{c};\ \mathrm{PP}\end{array}}{\mathsf{gave}\circ\varphi\circ\mathsf{to}\circ\mathsf{chris};\ \mathbf{gave}(x)(\mathbf{c});\ \mathrm{NP}\backslash\mathrm{S}}\,/\mathrm{E}}{\mathsf{kim}\circ\mathsf{gave}\circ\varphi\circ\mathsf{to}\circ\mathsf{chris};\ \mathbf{gave}(x)(\mathbf{c})(\mathbf{k});\ \mathrm{S}}\,\backslash\mathrm{E}}{\lambda\varphi.\mathsf{kim}\circ\mathsf{gave}\circ\varphi\circ\mathsf{to}\circ\mathsf{chris};\ \lambda x.\mathbf{gave}(x)(\mathbf{c})(\mathbf{k});\ \mathrm{S}{\uparrow}\mathrm{NP}}\,{\uparrow}\mathrm{I}^1}\,{\uparrow}\mathrm{E}}{\lambda\varphi.\varphi\circ\mathsf{kim}\circ\mathsf{gave}\circ\mathsf{to}\circ\mathsf{chris};\ \lambda x.\mathbf{gave}(x)(\mathbf{c})(\mathbf{k});\ \mathrm{S}{\uparrow}\mathrm{NP}}\,{\uparrow}\mathrm{E}}{\mathsf{bagels}\circ\mathsf{kim}\circ\mathsf{gave}\circ\mathsf{to}\circ\mathsf{chris};\ \mathbf{gave}(\mathbf{b})(\mathbf{c})(\mathbf{k});\ \mathrm{S}}\,{\uparrow}\mathrm{E}$$

In (17), a gapped sentence is derived just in the same way as in the quantifier example above via hypothetical reasoning for $\uparrow$ (①). The difference from the quantifier example is that the topicalization operator embeds an empty string $\boldsymbol{\epsilon}$ to the gap position, thereby closing off the gap, and then concatenates the string of the topicalized NP to the left of that string.

In short, what we have here can be thought of as a logical reconceptualization of transformational grammar; what we used to call 'trace' in derivational approaches is just a hypothesis in a proof and 'movement' is just a fancy name for a particular type of hypothetical reasoning.[5] This key insight, which seems to stem from the early 90s when the natural deduction formulations of various extensions of the Lambek calculus were introduced (see, e.g., Hepple (1990); Morrill (1994)) and which is expressed particularly clearly in Oehrle (1994), is not only theoretically illuminating, capturing the tight correlation between the semantic and prosodic effects of quantification and extraction transparently, but it also has a number of empirical advantages as well. Specifically, this approach enables an explicit and precise characterization of more complex types of scope-taking such as PARASITIC SCOPE of symmetrical predicates (Barker 2007; Pollard and Smith 2012) and SPLIT SCOPE of negative quantifiers (Kubota and Levine 2014a) and number determiners (Pollard 2014). Gapping is

[5]But caution should be taken that this is just a rough and crude analogy. In fact, one important property of Slash Introduction rules (including the Vertical Slash Introduction rule above) is that, as inference rules in the logical deductive system, they *define* the properties of the slashes together with the Slash Elimination rules. For this reason, the grammar rules in (TL)CG have very different statuses conceptually from 'corresponding' rules in other theories.

especially interesting in this connection in that it exhibits the properties of both 'overt' and 'covert' movement simultaneously, thus constituting a case that goes beyond the analytic possibilities available in the standard derivational architecture. We illustrate these points in detail in later chapters.

3.4 Hypothetical reasoning for all slashes: Hybrid Type-Logical Categorial Grammar

At this point, we extend our fragment once more, this time by adding the Introduction rules for the forward and backward slashes. This gives us the full Hybrid TLCG, complete with both the Introduction and Elimination rules for all the three slashes /, \ and ⇂. The main motivation for extending the system with the Introduction rules for the directional slashes comes from the analysis of coordination—including cases of nonconstituent coordination (Right-Node Raising (RNR) and Dependent Cluster Coordination (DCC)), as we illustrate below.

The Slash Introduction rules for / and \ are formulated as follows:

(18) a. Forward Slash Introduction

$$\dfrac{\begin{array}{c}\vdots\quad\vdots\quad\underline{[\varphi;\ x;\ A]^n}\quad\vdots\quad\vdots\\ \vdots\quad\vdots\qquad\vdots\quad\vdots\qquad\vdots\quad\vdots\end{array}}{\dfrac{b\circ\varphi;\ \boldsymbol{\mathcal{F}};\ B}{b;\ \lambda x.\boldsymbol{\mathcal{F}};\ B/A}\,/\mathrm{I}^n}$$

b. Backward Slash Introduction

$$\dfrac{\begin{array}{c}\vdots\quad\vdots\quad\underline{[\varphi;\ x;\ A]^n}\quad\vdots\quad\vdots\\ \vdots\quad\vdots\qquad\vdots\quad\vdots\qquad\vdots\quad\vdots\end{array}}{\dfrac{\varphi\circ b;\ \boldsymbol{\mathcal{F}};\ B}{b;\ \lambda x.\boldsymbol{\mathcal{F}};\ A\backslash B}\,\backslash\mathrm{I}^n}$$

The difference between the Introduction rule for the vertical slash and the Introduction rules for the directional slashes is that, unlike in ⇂I, in the /I and \I rules, the prosodic variable φ for the hypothesis is simply thrown away in the conclusion on the condition of its presence at the (either right or left) periphery of the prosody of the premise, instead of explicitly being bound by the λ-operator. The position of the missing expression is instead recorded in the forward vs. backward slash distinction in the syntactic category.

As will become clear in a moment, with the Introduction rules for / and \, it becomes possible to reanalyze any substring of a sentence as a (derived) constituent. We first illustrate the workings of these rules with analyses of RNR and DCC, and then come back to the relevant formal details. (20) shows how the string *John loves* in the RNR example in (19) is assigned the syntactic category S/NP via hypothetical reasoning involving /.

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

(20)

$$\textcircled{2}\rightarrow\dfrac{\mathsf{john};\ \mathbf{j};\ \mathrm{NP}\qquad\dfrac{\dfrac{[\varphi;x;\mathrm{NP}]^1\qquad\mathsf{loves};\ \mathbf{love};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}}{\mathsf{loves}\circ\varphi;\ \mathbf{love}(x);\ \mathrm{NP}\backslash\mathrm{S}}\,\backslash\mathrm{E}}{\dfrac{\mathsf{john}\circ\mathsf{loves}\circ\varphi;\ \mathbf{love}(x)(\mathbf{j});\ \mathrm{S}}{\mathsf{john}\circ\mathsf{loves};\ \lambda x.\mathbf{love}(x)(\mathbf{j});\ \mathrm{S}/\mathrm{NP}}\,/\mathrm{I}^1}\,/\mathrm{E}$$

By hypothesizing a direct object NP, we first prove an S (①). At this point, since the prosody of the hypothesized NP φ appears on the right periphery, we can apply /I to

withdraw the hypothesis (②). Intuitively, what is going on here can be paraphrased as follows: since we've proven that there is a complete S by assuming that there is an NP on the right periphery (①), we know that, without this hypothetical NP, what we have is something that becomes an S *if* there is an NP to its right (②). Thus, this is another instance of hypothetical reasoning. The difference from the case for $\uparrow$ is that, instead of explicitly keeping track of the position of the missing expression via λ-binding in the prosodic representation, the prosodic variable φ is simply thrown away, and the slash in the syntactic category records the fact that the NP is missing on the right periphery. Note also that the lambda abstraction on the corresponding variable in semantics assigns the right meaning (of type $e \to t$) to the derived S/NP.

In the CG analysis of RNR (see, e.g., Morrill (1994); the original analytic insight goes back to Steedman (1985)) such nonconstituents are directly coordinated as constituents and then combined with the RNR'ed expression as in (21):

(21)
$$\dfrac{\dfrac{\dfrac{\vdots\quad\vdots}{\mathsf{john} \circ \mathsf{loves};\ \lambda x.\mathbf{love}(x)(\mathbf{j});\ \mathrm{S/NP}} \qquad \dfrac{\mathsf{and};\ \lambda\mathcal{W}\lambda\mathcal{V}.\mathcal{V} \sqcap \mathcal{W};\ (X\backslash X)/X \qquad \dfrac{\vdots\quad\vdots}{\mathsf{bill} \circ \mathsf{hates};\ \lambda x.\mathbf{hate}(x)(\mathbf{b});\ \mathrm{S/NP}}\,/\mathrm{E}}{\mathsf{and} \circ \mathsf{bill} \circ \mathsf{hates};\ \lambda\mathcal{V}.\mathcal{V} \sqcap \lambda x.\mathbf{hate}(x)(\mathbf{b});\ (\mathrm{S/NP})\backslash(\mathrm{S/NP})}\,\backslash\mathrm{E}}{\mathsf{john} \circ \mathsf{loves} \circ \mathsf{and} \circ \mathsf{bill} \circ \mathsf{hates};\ \lambda x.\mathbf{love}(x)(\mathbf{j}) \sqcap \lambda x.\mathbf{hate}(x)(\mathbf{b});\ \mathrm{S/NP} \qquad \begin{array}{l}\mathsf{mary};\\ \mathbf{m};\ \mathrm{NP}\end{array}}\,/\mathrm{E}$$
$$\mathsf{john} \circ \mathsf{loves} \circ \mathsf{and} \circ \mathsf{bill} \circ \mathsf{hates} \circ \mathsf{mary};\ \mathbf{love}(\mathbf{m})(\mathbf{j}) \wedge \mathbf{hate}(\mathbf{m})(\mathbf{b});\ \mathrm{S}$$

Note in particular that this analysis assigns the right meaning to the whole sentence compositionally. $\sqcap$ designates GENERALIZED CONJUNCTION (Partee and Rooth 1983), defined as follows:[6]

(22) a. For p and q of type t, $p \sqcap q =_{def} p \wedge q$

b. For $\mathcal{P}$ and $\mathcal{Q}$ of any conjoinable type other than t (where $\mathcal{P}$ and $\mathcal{Q}$ are of the same type), $\mathcal{P} \sqcap \mathcal{Q} =_{def} \lambda\mathcal{R}.\mathcal{P}(\mathcal{R}) \sqcap \mathcal{Q}(\mathcal{R})$

Thus, $\lambda x.\mathbf{love}(x)(\mathbf{j}) \sqcap \lambda x.\mathbf{hate}(x)(\mathbf{b}) = \lambda x.\mathbf{love}(x)(\mathbf{j}) \wedge \mathbf{hate}(x)(\mathbf{b})$.

This analysis of RNR immediately extends to DCC. (23) shows the 'reanalysis' of the string *Bill the book* as a derived constituent. This involves first hypothesizing a ditransitive verb and withdrawing that hypothesis after a whole verb phrase is formed (here, VP abbreviates NP\S):

(23)
$$\dfrac{\dfrac{\dfrac{[\varphi; f; \mathrm{VP/NP/NP}]^1 \qquad \mathsf{bill};\ \mathbf{b};\ \mathrm{NP}}{\varphi \circ \mathsf{bill};\ f(\mathbf{b});\ \mathrm{VP/NP}}\,/\mathrm{E} \qquad \mathsf{the} \circ \mathsf{book};\ \iota(\mathbf{bk});\ \mathrm{NP}}{\varphi \circ \mathsf{bill} \circ \mathsf{the} \circ \mathsf{book};\ f(\mathbf{b})(\iota(\mathbf{bk}));\ \mathrm{VP}}\,/\mathrm{E}}{\mathsf{bill} \circ \mathsf{the} \circ \mathsf{book};\ \lambda f.f(\mathbf{b})(\iota(\mathbf{bk}));\ (\mathrm{VP/NP/NP})\backslash\mathrm{VP}}\,\backslash\mathrm{I}^1$$

[6] A conjoinable type is t and any functional type that 'bottoms in' t such as $e \to t$, $t \to t$, $et \to t$, $et \to et$, etc. Some more examples of generalized conjunction are as follows:

(i) a. $\mathbf{bought}_{et} \sqcap \mathbf{ate}_{et} \equiv \lambda x\lambda y.\mathbf{bought}(x)(y) \wedge \mathbf{ate}(x)(y)$

b. $\mathbf{yesterday}_{et\to et} \sqcap \mathbf{today}_{et\to et} \equiv \lambda P\lambda x.\mathbf{yesterday}(P)(x) \wedge \mathbf{today}(P)(x)$

Then, after like-category coordination, the missing verb and the subject NP are supplied to yield a complete sentence (the last step is omitted):

(24)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{gave};\ \mathbf{gave};\ \mathrm{VP/NP/NP}} \quad \dfrac{\dfrac{\dfrac{}{\mathsf{bill}\circ\mathsf{the}\circ\mathsf{book};\ \lambda f.f(\mathbf{b})(\iota(\mathbf{bk}));\ (\mathrm{VP/NP/NP})\backslash\mathrm{VP}} \quad \dfrac{\dfrac{}{\mathsf{and};\ \lambda\mathcal{W}\lambda\mathcal{V}.\mathcal{V}\sqcap\mathcal{W};\ (X\backslash X)/X} \quad \dfrac{}{\mathsf{john}\circ\mathsf{the}\circ\mathsf{record};\ \lambda f.f(\mathbf{j})(\iota(\mathbf{rc}));\ (\mathrm{VP/NP/NP})\backslash\mathrm{VP}}\,/\mathrm{E}}{\mathsf{and}\circ\mathsf{john}\circ\mathsf{the}\circ\mathsf{record};\ \lambda\mathcal{V}.\mathcal{V}\sqcap\lambda f.f(\mathbf{j})(\iota(\mathbf{rc}));\ ((\mathrm{VP/NP/NP})\backslash\mathrm{VP})\backslash((\mathrm{VP/NP/NP})\backslash\mathrm{VP})}\,\backslash\mathrm{E}}{\mathsf{bill}\circ\mathsf{the}\circ\mathsf{book}\circ\mathsf{and}\circ\mathsf{john}\circ\mathsf{the}\circ\mathsf{record};\ \lambda f.f(\mathbf{b})(\iota(\mathbf{bk}))\sqcap\lambda f.f(\mathbf{j})(\iota(\mathbf{rc}));\ (\mathrm{VP/NP/NP})\backslash\mathrm{VP}}\,\backslash\mathrm{E}}{\mathsf{gave}\circ\mathsf{bill}\circ\mathsf{the}\circ\mathsf{book}\circ\mathsf{and}\circ\mathsf{john}\circ\mathsf{the}\circ\mathsf{record};\ \mathbf{gave}(\mathbf{b})(\iota(\mathbf{bk}))\sqcap\mathbf{gave}(\mathbf{j})(\iota(\mathbf{rc}));\ \mathrm{VP}}\,\backslash\mathrm{E}$$

We now turn to some formal details about the Introduction rules in (18). First, since $\circ$ is string concatenation, if we remove the rules for the vertical slash (i.e. $\uparrow$I and $\uparrow$E) from the present fragment, the system is identical to the product-free (associative) Lambek calculus **L** (Lambek 1958). This means that, with the /I and \I rules, a hypothesis can be withdrawn as long as its prosody appears either on the left or right periphery of the prosody of the premise.[7] Note also that in this formulation, the prosodic term labelling, rather than the left-to-right order of the premises in the proof tree, is relevant for the applicability conditions of the /I and \I rules (the latter presentation is more common in the literature of mathematical linguistics; so far as we are aware, Morrill (1994) was the first to recast the Lambek calculus in the former format). This point should be clear from the proof in (20), where we have deliberately placed the hypothetical object NP to the *left* of the verb in the proof tree. This also means that the order of the two premises in the Elimination rules does not play any role, as we have already noted above. In practice, we often write premises in an order reflecting the actual word order, but it should be kept in mind that this is only for the sake of maintaining the readability of derivations.

3.5 Coordination and scope

As illustrated above, hypothetical reasoning with the directional slashes enables a simple and elegant analysis of coordination (including NCC). The real advantage of this approach, however, comes from a range of more complex examples involving interactions with scopal expressions. We illustrate this briefly here with examples involving generalized quantifiers.

[7] One might worry about overgeneration in this regard. For example, note that the infamous Dekker's puzzle arises in our setup, just as in the Lambek calculus, overgenerating the following string as a sentence:

(i) *[Bill thinks]$_{\mathrm{S/VP/NP}}$, and [the brother of]$_{\mathrm{S/VP/NP}}$, John walks.

The standard response to this issue in contemporary CG is to control the availability of associativity in the prosodic calculus by the notion of MULTI-MODALITY (Moortgat and Oehrle 1994; Dowty 1996; Baldridge 2002; Muskens 2007; Kubota 2010). This will, for example, prevent the inference S/S $\vdash$ S/VP/NP (valid in **L**) in the enriched calculus. Hybrid TLCG can be extended along these lines. See Kubota (2010, 2014) for a detailed study of word order-related phenomena found in the domain of complex predicates (including their interaction with coordination) exploiting this technique.

The issue will be discussed in greater detail with more complex types of scope-taking expressions (such as the 'respective' readings of conjoined and plural expressions and the internal readings of symmetrical predicates such as *same* and *different*) in chapter 5.

The case in point comes from examples like the following, discussed in chapter 2:

(25) Terry said nothing to Robin on Thursday or to Leslie on Friday.

Recall from chapter 2 the COORDINATION-SCOPE GENERALIZATION that crosscuts both the (syntactic and semantic) type of coordination involved and the type of scope-taking expression involved.[8] Basically, the generalization is very simple: when a scope-taking expression appears outside the coordinate structure in the surface string, both the WIDE SCOPE READING (in which the scopal operator scopes over the conjunction) and the DISTRIBUTIVE READING (in which the scope-taking expression scopes within each conjunct separately) are available. Which of the two readings is more prominent (or more readily available) in a particular sentence often depends heavily on pragmatic factors (in particular, the distributive reading tends to be more difficult in many examples), but there is reason to believe that the combinatoric component of grammar makes both $\Phi(\alpha \wedge \beta)$ and $\Phi(\alpha) \wedge \Phi(\beta)$ as possible readings for all sentences of the form 'Φ [$_{\mathrm{A}}$ [$_{\mathrm{A}}$ α] *and* [$_{\mathrm{A}}$ β]]'. Similarly for disjunction.

For (25), the more prominent reading is the wide scope reading for the quantifier, in which the negative quantifier scopes over the disjunction yielding an entailment of negation of two propositions (i.e. $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$). This wide scope reading poses a particularly challenging problem for certain types of analyses of NCC, as we have discussed in chapter 2. It turns out that, in Hybrid TLCG, the availability of both the wide scope reading and the distributive reading is automatically predicted from the independently motivated analyses of coordination and quantification already introduced above.

We illustrate the analysis of the quantifier wide scope reading first, since the derivation is somewhat simpler for this reading. The derivation is given in (26).

(26)

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

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

[8] This chapter is not yet written. See section 2 'The empirical domain: coordination and scope' in Kubota and Levine (2015a) for the relevant discussion.

$$
\dfrac{\begin{array}{l}\lambda\sigma.\sigma(\mathsf{nothing});\\ \neg\exists_{\mathbf{thing}};\\ \mathrm{S}{\upharpoonright}(\mathrm{S}{\upharpoonright}\mathrm{NP})\end{array} \quad \dfrac{\dfrac{\begin{array}{l}\mathsf{terry};\\ \mathbf{t};\mathrm{NP}\end{array} \quad \dfrac{\begin{array}{l}\mathsf{said};\\ \mathbf{said};\\ \mathrm{VP/NP/PP}\end{array} \quad \dfrac{\begin{bmatrix}\varphi_3;\\ x;\mathrm{NP}\end{bmatrix}^3 \quad \begin{array}{l}\vdots \quad \vdots\\ \mathsf{to \circ robin \circ on \circ thursday \circ or \circ to \circ leslie \circ on \circ 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(\mathrm{VP/PP/NP})\backslash\mathrm{VP}\end{array}}{\begin{array}{l}\varphi_3 \circ \mathsf{to \circ robin \circ on \circ thursday \circ or \circ to \circ leslie \circ on \circ friday};\\ \lambda P\lambda z.\mathbf{onTh}(P(x)(\mathbf{r}))(z) \vee \mathbf{onFr}(P(x)(\mathbf{l}))(z);\\ (\mathrm{VP/PP/NP})\backslash\mathrm{VP}\end{array}}\backslash \mathrm{E}}{\begin{array}{l}\mathsf{said} \circ \varphi_3 \circ \mathsf{to \circ robin \circ on \circ thursday \circ or \circ to \circ leslie \circ on \circ 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}}{\begin{array}{l}\mathsf{terry \circ said} \circ \varphi_3 \circ \mathsf{to \circ robin \circ on \circ thursday \circ or \circ to \circ leslie \circ on \circ friday};\\ \mathbf{onTh}(\mathbf{said}(x)(\mathbf{r}))(\mathbf{t})\mathbf{onFr}(\mathbf{said}(x)(\mathbf{l}))(\mathbf{t}); \mathrm{S}\end{array}}\backslash \mathrm{E}}{\begin{array}{l}\lambda\varphi_3.\mathsf{terry \circ said} \circ \varphi_3 \circ \mathsf{to \circ robin \circ on \circ thursday \circ or \circ to \circ leslie \circ on \circ 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}{\begin{array}{l}\mathsf{terry \circ said \circ nothing \circ to \circ robin \circ on \circ thursday \circ or \circ to \circ leslie \circ on \circ friday};\\ \neg\exists_{\mathbf{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.

The derivation for the distributive reading is a bit more complex. We first illustrate the relevant steps with the constituent coordination example (27) and then show that the NCC case is treated in essentially the same way.

(27) During the past month, a mob boss was assassinated in Boston and executed in New York.

To see how the distributive reading is derived, note first that a quantifier entry of type $\mathrm{S}{\upharpoonright}(\mathrm{S}{\upharpoonright}\mathrm{NP})$ (with prosodic type $(\mathbf{st} \to \mathbf{st}) \to \mathbf{st}$) can be converted to a sign with type $\mathrm{S}/(\mathrm{NP}\backslash\mathrm{S})$ (with a simple string prosody) via a series of inference steps as follows (see the next section for a step-by-step 'verbalization' of this proof):

(28)

$$
\dfrac{\dfrac{\lambda\sigma.\sigma(\mathsf{a \circ mob \circ boss}); \exists_{\mathbf{mb}}; \mathrm{S}{\upharpoonright}(\mathrm{S}{\upharpoonright}\mathrm{NP}) \quad \dfrac{\dfrac{[\varphi_1; x; \mathrm{NP}]^1 \quad [\varphi_2; P; \mathrm{NP}\backslash\mathrm{S}]^2}{\varphi_1 \circ \varphi_2;\ P(x);\ \mathrm{S}}\backslash \mathrm{E}}{\lambda\varphi_1.\varphi_1 \circ \varphi_2;\ \lambda x.P(x);\ \mathrm{S}{\upharpoonright}\mathrm{NP}}{\upharpoonright}\mathrm{I}^1}{\mathsf{a \circ mob \circ boss} \circ \varphi_2;\ \exists_{\mathbf{mb}}(\lambda x.P(x));\ \mathrm{S}}{\upharpoonright}\mathrm{E}}{\mathsf{a \circ mob \circ boss};\ \lambda P.\exists_{\mathbf{mb}}(\lambda x.P(x));\ \mathrm{S}/(\mathrm{NP}\backslash\mathrm{S})}/\mathrm{I}^2
$$

This 'lowered' quantifier entry is identical to the familiar type specification for quantifiers in variants of CG with directional slashes only such as CCG.

Once we have this derived entry, the distributive reading is straightforwardly obtained. The derivation involves first lifting the type of the VP to take GQ-type arguments in the subject position and then coordinating such higher-order VPs, to which the quantifier is then given as an argument:

(29)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\left[\begin{array}{l}\varphi;\\ \mathscr{P};\\ \mathrm{S/VP}\end{array}\right]^{1} \quad \begin{array}{l}\mathsf{was}\circ\mathsf{assassinated}\circ\\ \mathsf{in}\circ\mathsf{boston};\\ \mathbf{ass'd\text{-}in\text{-}b};\ \mathrm{VP}\end{array}}{\begin{array}{l}\varphi\circ\mathsf{was}\circ\mathsf{assassinated}\circ\\ \mathsf{in}\circ\mathsf{boston};\\ \mathscr{P}(\mathbf{ass'd\text{-}in\text{-}b});\ \mathrm{S}\end{array}}\backslash\mathrm{E} \quad \begin{array}{l}\mathsf{and};\\ \lambda\mathcal{W}\lambda\mathcal{V}.\mathcal{V}\sqcap\mathcal{W};\\ (X\backslash X)/X\end{array} \quad \begin{array}{l}\vdots\\ \mathsf{was}\circ\mathsf{executed}\circ\\ \mathsf{in}\circ\mathsf{ny};\\ \lambda\mathscr{P}.\mathscr{P}(\mathbf{ex'd\text{-}in\text{-}ny});\\ (\mathrm{S/VP})\backslash\mathrm{S}\end{array}}{\begin{array}{l}\mathsf{and}\circ\mathsf{was}\circ\mathsf{executed}\circ\mathsf{in}\circ\mathsf{ny};\\ \lambda\mathcal{V}.\mathcal{V}\sqcap\lambda\mathscr{P}.\mathscr{P}(\mathbf{ex'd\text{-}in\text{-}ny});\\ ((\mathrm{S/VP})\backslash\mathrm{S})\backslash((\mathrm{S/VP})\backslash\mathrm{S})\end{array}}/\mathrm{E}}{\begin{array}{l}\mathsf{was}\circ\mathsf{assassinated}\circ\mathsf{in}\circ\mathsf{boston}\circ\mathsf{and}\circ\mathsf{was}\circ\mathsf{executed}\circ\mathsf{in}\circ\mathsf{ny};\\ \lambda\mathscr{P}.\mathscr{P}(\mathbf{ass'd\text{-}in\text{-}b})\sqcap\lambda\mathscr{P}.\mathscr{P}(\mathbf{ex'd\text{-}in\text{-}ny});\ (\mathrm{S/VP})\backslash\mathrm{S}\end{array}}\backslash\mathrm{E}}{\begin{array}{l}\mathsf{a}\circ\mathsf{mob}\circ\mathsf{boss};\\ \mathfrak{Exists}_{\mathbf{mb}};\ \mathrm{S/VP}\end{array}}\backslash\mathrm{E}}{\begin{array}{l}\mathsf{a}\circ\mathsf{mob}\circ\mathsf{boss}\circ\mathsf{was}\circ\mathsf{assassinated}\circ\mathsf{in}\circ\mathsf{boston}\circ\mathsf{and}\circ\mathsf{was}\circ\mathsf{executed}\circ\mathsf{in}\circ\mathsf{ny};\\ \mathfrak{Exists}_{\mathbf{mb}}(\mathbf{ass'd\text{-}in\text{-}b})\wedge\mathfrak{Exists}_{\mathbf{mb}}(\mathbf{ex'd\text{-}in\text{-}ny});\ \mathrm{S}\end{array}}/\mathrm{I}^{1}$$

We call the type of derivation shown in (28) SLANTING, in the sense that an expression involving the vertical slash is converted to one involving 'slanted' (i.e. forward and backward) slashes. As in (28), slanting exploits the 'hybrid' architecture of the present framework in which syntactic inferences involving the vertical slash freely interact with those involving the forward and backward slashes. See the next section for more on slanting.

Note that the ambiguity for quantifiers in sentences involving coordination is due to the polymorphic specification for the conjunction word *and*. Without this lexical ambiguity for the conjunction word, slanting would just result in a detour in the proof without any semantic consequence and which will be systematically eliminated in proof normalization.

Since slanting and type-lifting are generally available as theorems for any argument position of a verb, they can be applied to induce the distributive reading for quantifiers that occupy any argument position of a verb (extension to adjunct positions is also straightforward). In the NCC example in (30), the quantifier occupies the direct object position of a prepositional ditransitive verb.

(30) Terry gave a present to Robin on Thursday and to Leslie on Friday.

Thus, the distributive reading for this quantifier can be obtained by lifting the type of this argument position for the verb and slanting the quantifier accordingly. Note that the type of the argument clusters is a bit more complex than in the simpler cases above since they are specified to take the slanted quantifier as one of their arguments (so that the quantifier meaning is distributed to each conjunct). The derivation is given in (31). (Here, PTV abbreviates VP/PP; note that we are assuming an 'already slanted' version of the quantifier in (31)—for the derivation of this (PTV/NP)\PTV entry for the quantifier, see (34) in the next section.)

(31)

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

$$
\dfrac{\text{gave; } \mathbf{gave}; \text{ PTV/NP} \qquad \dfrac{\begin{array}{l} \vdots \quad \vdots \\ \mathsf{a \circ present}; \\ \lambda P \lambda y \lambda z.\mathbf{\exists_{pr}}(\lambda x.P(x)(y)(z)); \\ \text{(PTV/NP)\textbackslash PTV} \end{array} \qquad \begin{array}{l} \vdots \quad \vdots \\ \mathsf{to \circ robin \circ on \circ thursday \circ} \\ \mathsf{and \circ to \circ leslie \circ on \circ friday}; \\ \lambda \mathscr{P} \lambda P.\mathbf{onTh}(\mathscr{P}(P)(\mathbf{r})) \sqcap \\ \lambda \mathscr{P} \lambda P.\mathbf{onFr}(\mathscr{P}(P)(\mathbf{l})); \\ \text{((PTV/NP)\textbackslash PTV)\textbackslash (PTV/NP)\textbackslash VP} \end{array}}{\begin{array}{l} \mathsf{a \circ present \circ to \circ robin \circ on \circ thursday \circ} \\ \mathsf{and \circ to \circ leslie \circ on \circ friday}; \\ \lambda P.\mathbf{onTh}(\lambda z.\mathbf{\exists_{pr}}(\lambda x.P(x)(\mathbf{r})(z))) \sqcap \lambda P.\mathbf{onFr}(\lambda z.\mathbf{\exists_{pr}}(\lambda x.P(x)(\mathbf{l})(z))); \\ \text{(PTV/NP)\textbackslash VP} \end{array}} \backslash \text{E}}{\begin{array}{l} \mathsf{gave \circ a \circ present \circ to \circ robin \circ on \circ thursday \circ and \circ to \circ leslie \circ on \circ friday}; \\ \mathbf{onTh}(\lambda z.\mathbf{\exists_{pr}}(\lambda x.\mathbf{gave}(x)(\mathbf{r})(z))) \sqcap \mathbf{onFr}(\lambda z.\mathbf{\exists_{pr}}(\lambda x.\mathbf{gave}(x)(\mathbf{l})(z))); \text{VP} \end{array}} \backslash \text{E}
$$

Thus, the present approach licenses both the distributive and non-distributive readings for quantifiers when they interact with coordination, both in the constituent coordination and NCC cases. We take this to be an empirically correct result. As we discussed in chapter 2, the apparent difficulty for the distributive reading for downward entailing quantifiers disappears once appropriate contexts are established.[9]

3.6 Slanting

As discussed in Sect. 3.5, an operation (or a family of theorems) called SLANTING plays a crucial role in deriving entries of quantifiers that are used in licensing distributive readings of quantifiers in coordination examples. We reproduce here the slanting derivation for the subject position quantifier (32) (= (28)), together with two more cases, one for the object position for transitive verbs (33) and the other for the direct object position of prepositional ditransitive verbs (34) (the latter is used in the derivation of the distributive reading for an NCC example in (31)).

Since slanting is a (set of) lemma applicable to a set of lexical entries rather than just to some specific words, we present it in a schematic form with quantifier entries with some

[9]There does nonetheless seem to be an overwhelming preference for the non-distributive readings for downward entailing quantifiers in many cases, especially in ‘out of the blue’ contexts. We believe that this is a reflection of a much larger generalization about the (preferred) scopal relation between negation (which is part of the meanings of downward entailing quantifiers) and other operators. Specifically, negation tends to resist inverse scope readings in relation to operators that it ‘c-commands’:

(i) a. John didn’t talk to every teacher. (??$\forall > \neg$)
 b. No student/few students/hardly any student talked to every teacher. (??$\forall >$ *no, few, hardly any*)

Note in particular that ordinary negation exhibits basically the same pattern as downward entailing quantifiers when it interacts with coordination:

(ii) a. Terry didn’t say anything to Robin or Leslie. ($\neg > \vee$, *$\vee > \neg$)
 b. Terry didn’t say anything to Robin on Thursday or to Leslie on Friday. ($\neg > \vee$, *$\vee > \neg$)

At least in the disjunction cases like (ii), the dispreference for the distributive reading is clear in both constituent coordination and NCC. The relative inaccessibility of the distributive reading in such examples most likely results from a complex interaction of multiple factors, one of which is the absence (in ordinary contexts) of the relevant discourse relations (of the sort discussed in chapter 2) supporting the distributive reading.

string prosody p, where p is a metavariable ranging over the set of actual quantifier strings. We start with slanting for the subject position quantifier:

(32)
$$\dfrac{\lambda\sigma.\sigma(p);\ \mathfrak{A}_{\mathbf{mb}};\ \mathrm{S}{\uparrow}(\mathrm{S}{\uparrow}\mathrm{NP}) \quad \dfrac{\dfrac{[\varphi_1; x; \mathrm{NP}]^1 \quad [\varphi_2; P; \mathrm{NP}\backslash\mathrm{S}]^2}{\varphi_1 \circ \varphi_2;\ P(x);\ \mathrm{S}}\,\backslash\mathrm{E}}{\lambda\varphi_1.\varphi_1 \circ \varphi_2;\ \lambda x.P(x);\ \mathrm{S}{\uparrow}\mathrm{NP}}\,{\uparrow}\mathrm{I}^1}{\dfrac{p \circ \varphi_2;\ \mathfrak{A}_{\mathbf{mb}}(\lambda x.P(x));\ \mathrm{S}}{p;\ \lambda P.\mathfrak{A}_{\mathbf{mb}}(\lambda x.P(x));\ \mathrm{S}/(\mathrm{NP}\backslash\mathrm{S})}\,/\mathrm{I}^2}\,{\uparrow}\mathrm{E}$$

The derivation in (32) can be understood as follows. We start by hypothesizing a VP (i.e., NP\S, indexed as 2) and an NP (indexed as 1). After a complete sentence is formed, the NP hypothesis is withdrawn so that the quantifier can take scope. The resulting expression is indeed of the right type (S↑NP) to be given as an argument to the quantifier, and the quantifier string is lowered to the subject position. This yields a string in which the prosody of the hypothesized VP φ_2 appears at the right periphery, satisfying the applicability condition of /I. By applying /I, the quantifier string p is paired with the original denotation of the quantifier (note that the final translation is equivalent to $\mathfrak{A}_{\mathbf{mb}}$ via eta-equivalence) and the syntactic category S/(NP\S).

Similar steps of derivation yield alternative quantifier entries in other argument positions of the sentence, as follows:

(33)
$$\dfrac{\lambda\sigma.\sigma(p);\ \mathfrak{A}_{\mathbf{mb}};\ \mathrm{S}{\uparrow}(\mathrm{S}{\uparrow}\mathrm{NP}) \quad \dfrac{\dfrac{[\varphi_2; P; \mathrm{S}/\mathrm{NP}]^2 \quad [\varphi_1; x; \mathrm{NP}]^1}{\varphi_2 \circ \varphi_1;\ P(x);\ \mathrm{S}}\,\backslash\mathrm{E}}{\lambda\varphi_1.\varphi_2 \circ \varphi_1;\ \lambda x.P(x);\ \mathrm{S}{\uparrow}\mathrm{NP}}\,{\uparrow}\mathrm{I}^1}{\dfrac{\varphi_2 \circ p;\ \mathfrak{A}_{\mathbf{mb}}(\lambda x.P(x));\ \mathrm{S}}{p;\ \lambda P.\mathfrak{A}_{\mathbf{mb}}(\lambda x.P(x));\ (\mathrm{S}/\mathrm{NP})\backslash\mathrm{S}}\,/\mathrm{I}^2}\,{\uparrow}\mathrm{E}$$

(34)
$$\dfrac{\dfrac{\lambda\sigma.\sigma(p);\ \mathfrak{A}_{\mathbf{pr}};\ \mathrm{S}{\uparrow}(\mathrm{S}{\uparrow}\mathrm{NP}) \quad \dfrac{\dfrac{\begin{bmatrix}\varphi_3;\\ z;\mathrm{NP}\end{bmatrix}^3 \quad \dfrac{\dfrac{\begin{bmatrix}\varphi_0;\\ P;\mathrm{VP/PP/NP}\end{bmatrix}^0 \quad \begin{bmatrix}\varphi_1;\\ x;\mathrm{NP}\end{bmatrix}^1}{\varphi_0 \circ \varphi_1;\ P(x);\ \mathrm{VP/PP}}\,/\mathrm{E} \quad \begin{bmatrix}\varphi_2;\\ y;\mathrm{PP}\end{bmatrix}^2}{\varphi_0 \circ \varphi_1 \circ \varphi_2;\ P(x)(y);\ \mathrm{VP}}\,/\mathrm{E}}{\varphi_3 \circ \varphi_0 \circ \varphi_1 \circ \varphi_2;\ P(x)(y)(z);\ \mathrm{S}}\,\backslash\mathrm{E}}{\lambda\varphi_1.\varphi_3 \circ \varphi_0 \circ \varphi_1 \circ \varphi_2;\ \lambda x.P(x)(y)(z);\ \mathrm{S}{\uparrow}\mathrm{NP}}\,{\uparrow}\mathrm{I}^1}{\dfrac{\dfrac{\dfrac{\varphi_3 \circ \varphi_0 \circ p \circ \varphi_2;\ \mathfrak{A}_{\mathbf{pr}}(\lambda x.P(x)(y)(z));\ \mathrm{S}}{\varphi_0 \circ p \circ \varphi_2;\ \lambda z.\mathfrak{A}_{\mathbf{pr}}(\lambda x.P(x)(y)(z));\ \mathrm{VP}}\,\backslash\mathrm{I}^3}{\varphi_0 \circ p;\ \lambda y\lambda z.\mathfrak{A}_{\mathbf{pr}}(\lambda x.P(x)(y)(z));\ \mathrm{PTV}}\,/\mathrm{I}^2}{p;\ \lambda P\lambda y\lambda z.\mathfrak{A}_{\mathbf{pr}}(\lambda x.P(x)(y)(z));\ (\mathrm{PTV}/\mathrm{NP})\backslash\mathrm{PTV}}\,\backslash\mathrm{I}^0}\,{\uparrow}\mathrm{E}$$

More generally, slanting is possible for any sentence-medial position. With S$ abbreviating any arbitrary directional syntactic type ending in S (a notation we borrow from Steedman (2000), where two occurrences of S$ with the same subscript instantiates the same category), slanting for quantifiers can be further schematized as follows:

(35) $\lambda\sigma.\sigma(p);\ \mathscr{P};\ \mathrm{S}{\uparrow}(\mathrm{S}{\uparrow}\mathrm{NP}) \ \vdash\ p;\ \lambda P\lambda x_1 \ldots x_n.\mathscr{P}(\lambda x.P(x)(x_1)\ldots(x_n));\ \mathrm{S}\$_1/(\mathrm{NP}\backslash\mathrm{S}\$_1)$

(36) $\lambda\sigma.\sigma(p);\ \mathscr{P};\ \mathrm{S}{\uparrow}(\mathrm{S}{\uparrow}\mathrm{NP}) \ \vdash\ p;\ \lambda P\lambda x_1 \ldots x_n.\mathscr{P}(\lambda x.P(x)(x_1)\ldots(x_n));\ (\mathrm{S}\$_2/\mathrm{NP})\backslash\mathrm{S}\$_2$

It is important to note that, although Slanting is a very flexible operation, it never yields entries of quantifiers that disrupt word order. For example, it is impossible to derive an entry for the quantifier of type (NP\S)\S from the lexically specified type S↑(S↑NP). (If such a derivation were possible, it would overgenerate the string **Loves a sailor every boy* with the reading 'Every boy loves a sailor', clearly a wrong result.)

The following failed derivation for S↑(S↑NP) ⊢ (NP\S)\S illustrates the point succinctly:

(37)
$$\dfrac{\lambda\sigma.\sigma(p);\ \exists\mathbf{mb};\ \mathrm{S}\uparrow(\mathrm{S}\uparrow\mathrm{NP}) \quad \dfrac{\dfrac{[\varphi_1; x; \mathrm{NP}]^1 \quad [\varphi_2; P; \mathrm{NP}\backslash\mathrm{S}]^2}{\varphi_1\circ\varphi_2;\ P(x);\ \mathrm{S}}\backslash\mathrm{E}}{\lambda\varphi_1.\varphi_1\circ\varphi_2;\ \lambda x.P(x);\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{I}^1}{\dfrac{p\circ\varphi_2;\ \exists\mathbf{mb}(\lambda x.P(x));\ \mathrm{S}}{\mathrm{FAIL}}**\backslash\mathrm{I}^{2}**}\uparrow\mathrm{E}$$

The proof starts by hypothesizing NP\S, since this is the category that needs to be withdrawn at the last step via \I to obtain the category (NP\S)\S. Then the proof proceeds in the same way as (32) above until the very final step, at which point it fails. At this final step, in order to obtain the (NP\S)\S category instead of the S/(NP\S) category as in (32), we need to apply \I. However, this is impossible since the applicability condition of the \I rule requires the prosody of the hypothesis to be withdrawn to appear on the left periphery of the input string—a condition that is not satisfied in (37). Thus, this derivation is correctly blocked.

Interestingly, unlike for quantifiers, verb lexical entries can be slanted and *un*slanted back and forth. The following derivations for a transitive verb illustrates this point:

(38)
$$\dfrac{\dfrac{\dfrac{[\varphi_2; y; \mathrm{NP}]^2 \quad \dfrac{p;\ R;\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP} \quad [\varphi_1; x; \mathrm{NP}]^1}{p\circ\varphi_1;\ R(x);\ \mathrm{NP}\backslash\mathrm{S}}/\mathrm{E}}{\varphi_2\circ p\circ\varphi_1;\ R(x)(y);\ \mathrm{S}}\backslash\mathrm{E}}{\lambda\varphi_2.\varphi_2\circ p\circ\varphi_1;\ \lambda y.R(x)(y);\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{I}^2}{\lambda\varphi_1\lambda\varphi_2.\varphi_2\circ p\circ\varphi_1;\ \lambda x\lambda y.R(x)(y);\ (\mathrm{S}\uparrow\mathrm{NP})\uparrow\mathrm{NP}}\uparrow\mathrm{I}^1$$

(39)
$$\dfrac{\dfrac{\dfrac{[\varphi_2; y; \mathrm{NP}]^2 \quad \dfrac{\lambda\varphi_3\lambda\varphi_4.\varphi_4\circ p\circ\varphi_3;\ R;\ (\mathrm{S}\uparrow\mathrm{NP})\uparrow\mathrm{NP} \quad [\varphi_1; x; \mathrm{NP}]^1}{\lambda\varphi_4.\varphi_4\circ p\circ\varphi_1;\ R(x);\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{E}}{\varphi_2\circ p\circ\varphi_1;\ R(x)(y);\ \mathrm{S}}\uparrow\mathrm{E}}{p\circ\varphi_1;\ \lambda y.R(x)(y);\ \mathrm{NP}\backslash\mathrm{S}}\backslash\mathrm{I}^2}{p;\ \lambda x\lambda y.R(x)(y);\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}}/\mathrm{I}^1$$

Slanting and unslanting that disrupt word order are underivable in Hybrid TLCG since the prosodic labelling keeps track of the string positions of both hypothesized and real expressions. That is, inferences that go against the linear order properties encoded via / and \ in the syntactic categories of the relevant expressions will be automatically ruled out in the calculus. This was illustrated for the failed quantifier slanting in (37). Similarly, the unslanted category S↑NP↑NP for the transitive verb obtained in the derivation in (38) does not give rise to any additional word order possibilities, since the word order information originally encoded in the / vs. \ distinction in the syntactic category is 'transferred' to the explicit encoding of word order in the (functional) prosodic specification of the derived expression. In this sense, proofs in the present calculus are strictly order-preserving.

3.7 A note on the linearity of the calculus

Since the treatment of quantification via prosodic λ-binding in the present framework is very powerful and flexible, one might worry about potential overgeneration of the following kind. Suppose we hypothesize the same variable in the two conjuncts of a conjoined sentence and bind them at once after the conjoined sentence is formed:

(40) $\lambda\varphi.\varphi \circ \mathsf{is} \circ \mathsf{male} \circ \mathsf{or} \circ \varphi \circ \mathsf{is} \circ \mathsf{female};\ \lambda x.\mathbf{male}(x) \vee \mathbf{female}(x);\ \mathrm{S}\uparrow\mathrm{NP}$

Then, by lowering the quantifier *everyone*, we seem to obtain the following:

(41) $\mathsf{everyone} \circ \mathsf{is} \circ \mathsf{male} \circ \mathsf{or} \circ \mathsf{everyone} \circ \mathsf{is} \circ \mathsf{female};\ \mathbf{\forall_{person}}(\lambda x.\mathbf{male}(x) \vee \mathbf{female}(x));\ \mathrm{S}$

In other words, we (apparently) incorrectly predict that *Everyone is male or everyone is female* has the reading 'everyone is either male or female'.

Such a derivation actually doesn't go through. This is ensured by the *linearity* of the vertical slash. That is, slashes in CG are modes of implication in a resource-sensitive logic, and this means that the three connectives /, \ and $\uparrow$ can bind only one occurrence of a hypothesis at a time. The basic underlying linguistic intuition is that one token of a linguistic expression can fill in only one argument position of a subcategorizing predicate. The derivation above, in which one token of the quantifier *everyone* fills in the subject argument position of two distinct verbs simultaneously is a textbook example of violation of resource sensitivity and hence is ruled out in the present framework (as in any other variant of CG).[10]

This of course raises the question of how to treat ATB extraction and parasitic gaps:

(42) a. John met a man who Mary likes __ but Sue hates __.
b. Which paper did John file __ without reading __?

This is indeed a non-trivial problem. We will address the issue of multiple gaps in chapter 8.

3.8 Comparison with related approaches

We end this chapter with a brief comparison of our framework with related approaches. Our discussion here should be useful for readers who have seen other variants of CG (such as CCG) and are at this point wondering about the relationship between our approach and these alternatives. Readers who have not encountered other variants of CG may want to skip this section for now and move on to the next chapter.

As should already be clear from the foregoing discussion, Hybrid TLCG is unlike other variants of contemporary CG in its emphasis on the similarities to the currently standard

[10] This does not exclude a possibility in which a non-linear semantic term is assigned as the translation for some linguistic expression (e.g. $\mathbf{\forall_{man}}(\lambda x.\mathbf{love}(x)(x))$ for *Every man loves himself*). This is possible in TLCG since non-linear terms can be introduced as the translations of specific lexical items (such as reflexives). By the same token, nothing blocks the non-linear term $\mathbf{\forall_{person}}(\lambda x.\mathbf{male}(x) \vee \mathbf{female}(x))$ from being assigned as the translation for *Everyone is either male or female.* Here, the generalized conjunction meaning for *and* is the source of non-linearity.

derivational architecture of grammar (and our presentation above has deliberately exaggerated this aspect).[11] This may appear puzzling to some, since CG has traditionally been regarded as belonging to the family of 'nonderivational' approaches to syntax.[12] We would therefore like to clarify below what we take to be the key similarities and differences between our approach and these related variants of CG.

Our discussion below focuses on two variants of CG: CCG and LCG. CCG is arguably a version of CG that is best known to linguists (and is clearly a 'nonderivational' variant of generative grammar), and LCG is important for us since we adopt the prosodic λ-binding treatment of quantification from it. But before getting into the main discussion, we want to briefly comment on the relationship between our approach and other variants of TLCG. Variants of TLCG can all be seen as extensions of the Lambek calculus to cope with its shortcomings in dealing with phenomena such as medial extraction and quantification. In the approach advocated by Moortgat (1997), Bernardi (2002) and Vermaat (2005), this is achieved by employing a technique from substructural logic for recognizing different kinds of logic (tied to distinct sets of 'structural' operations) simultaneously within a single calculus. The other line of work in TLCG, explored by Glyn Morrill and his colleagues (Morrill and Solias 1993, Morrill 1994 and Morrill et al. 2011, among others) recognizes a purpose-specific calculus for the prosodic component for dealing with discontinuity (in a broader sense encompassing extraction and quantification). While sharing the same general goals, we depart from these two traditions in one important way: our approach employs the standard λ-calculus for the prosodic component to deal with the discontinuity problem, thereby simplifying both the conceptual and technical foundations of TLCG considerably. Empirically, Hybrid TLCG is most closely related to Barker's (2007) and Barker and Shan's (2015) NL_λ. NL_λ is a version of the first type of TLCG described above utilizing structural postulates. The main differences between our calculus and NL_λ are that (i) NL_λ recognizes a novel structural postulate called the 'λ' postulate, whose conceptual and formal underpinnings are somewhat unclear (see Kubota (2010, 153, footnote 108) for some discussion on this point; but see also Barker and Shan (2015, Ch. 17) for a proof that NL_λ is equivalent to a slightly different calculus in which the 'λ' postulate is replaced by three distinct, 'perfectly kosher' structural rules in standard TLCG), whereas we achieve (more or less) the same effect with a completely standard λ-calculus (but our approach is more powerful in recognizing higher-order variables in the prosodic component; cf. Kubota and Levine (2013b)); and (ii) NL_λ, as its name suggests, is based on a *non-associative* variant of the Lambek calculus, whereas Hybrid TLCG is associative (though this difference is perhaps not so critical, since, as Barker and Shan (2015) note, introducing associativity in their system is straightforward; likewise, restricting associativity in our calculus is also straightforward—see Kubota (2014) for a concrete proposal along these lines).

[11] But it should also be kept in mind that there are many important differences between our approach and the standard derivational framework both conceptually and empirically, as we have indicated at various points above; we will elaborate further on these issues in due course in the ensuing chapters.

[12] Evidence for this can be found, for example, in Borsley and Börjars's (2011) handbook on 'Non-transformational syntax', which contains chapters on 'Combinatory Categorial Grammar' and 'Multi-Modal Type-Logical Grammar', two representative contemporary variants of CG.

3.8.1 Combinatory Categorial Grammar

CCG (Steedman 1996, 2000) is an extension of the AB grammar that proposes to simulate movement with a limited set of additional rules. Though there are several different variants and extensions, CCG typically consists of rules of TYPE RAISING and FUNCTION COMPOSITION, together with function application. Thus, the following represents a reasonable rule set for a simple CCG fragment (to facilitate comparison, we have written these rules in the tripartite sign format and have adopted the Lambek slash notation):[13]

(43) a. Forward Function Application

$$\frac{a;\ \boldsymbol{\mathcal{F}};\ A/B \qquad b;\ \boldsymbol{\mathcal{G}};\ B}{a \circ b;\ \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\ A}\ \text{FA}$$

b. Backward Function Application

$$\frac{b;\ \boldsymbol{\mathcal{G}};\ B \qquad a;\ \boldsymbol{\mathcal{F}};\ B\backslash A}{b \circ a;\ \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\ A}\ \text{FA}$$

(44) a. Forward Function Composition

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

b. Backward Function Composition

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

(45) a. Forward Type Raising

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

b. Backward Type Raising

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

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:

(46)

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

Thus, in CCG, essentially the same analysis of RNR is possible as in our approach. The same is true for DCC, as first noted by Dowty (1988). For details, we refer the reader to Dowty (1988) and Steedman (2000). Note, however, that in CCG the derived constituents in RNR and DCC are obtained by the rules in (43)–(45) rather than via hypothetical reasoning. In fact, CCG is notable for its strong thesis of 'surface compositionality', according to which the semantic interpretation of a sentence is directly obtained on the basis of the surface constituent structure assigned on the basis of a limited set of 'combinatory' rules like those in (43)–(45). In particular, there is no direct analog of vertical slash in CCG.

As far as the coverage of the basic syntactic patterns of NCC is concerned, CCG and Hybrid TLCG are more or less equivalent. But things are not so simple when we consider

[13] But it should be noted that the fragment here is an impoverished version of CCG for an expository purpose only. A linguistically more adequate version typically involves rules of 'crossed' function composition (for extraction from non-peripheral positions; cf. Steedman 1996) and the so-called 'substitution' rules for the treatment of parasitic gaps (Steedman 1987).

a wider range of data. First of all, CCG does not have a fully general analysis of the interactions between coordination and quantifier scope. An analysis of wide scope readings of quantifiers in coordination (including RNR and DCC) in examples like (47) is straightforward in CCG (as demonstrated in (48)).

(47) (Either) the department owns, or the library has an interlibrary license to, every single book in the SLAP series.

(48)
$$\frac{\begin{array}{ll}\vdots & \vdots \\ \mathsf{the} \circ \mathsf{department} \circ \mathsf{owns} \circ \mathsf{or} \circ \mathsf{the} \circ \mathsf{library} \circ \mathsf{has} \circ \mathsf{a} \circ \mathsf{license} \circ \mathsf{to};\ \mathbf{own}(\mathbf{d}) \sqcup \mathbf{has\text{-}license}(\mathbf{l}); \mathrm{S/NP} & \mathsf{every} \circ \mathsf{book};\ \forall_{\mathbf{book}}; (\mathrm{S/NP})\backslash\mathrm{S}\end{array}}{\mathsf{the} \circ \mathsf{department} \circ \mathsf{owns} \circ \mathsf{or} \circ \mathsf{the} \circ \mathsf{library} \circ \mathsf{has} \circ \mathsf{a} \circ \mathsf{license} \circ \mathsf{to} \circ \mathsf{every} \circ \mathsf{book};\ \forall_{\mathbf{book}}(\mathbf{own}(\mathbf{d}) \sqcup \mathbf{has\text{-}license}(\mathbf{l})); \mathrm{S}}\ \mathrm{FA}$$

However, it is unclear how the distributive readings of quantifiers, exemplified by data such as (49) (in its $\vee > \forall$ reading), are obtained in CCG.

(49) (The department used to have a heavy requirement for candidacy. For example, I no longer remember which it was, but) back in those days, every student had to pass at least two language exams or had to write QPs in both syntax and phonology.

In CCG, coordinated VPs are unambiguously of type NP\S and the subject quantifier has category S/(NP\S). Thus, only the weaker $\forall > \vee$ reading is predicted for this sentence. The discussion of the interaction between coordination and scope in Steedman (2012, Sect. 2.3; Ch. 10) in fact suggests that Steedman takes the distributive readings of quantifiers in coordination to be unavailable.[14] However, Kubota and Levine (2015a) show that distributive readings for quantifiers are in fact available for both constituent coordination and NCC, by setting up the appropriate pragmatic context and pronouncing the sentence in the right prosody.

It is of course conceivable to extend the rule set of CCG so that the $\vee > \forall$ reading is obtained for (49). In particular, what is needed here is the rule of ARGUMENT RAISING (Hendriks 1993), with which one can derive the type (S/(NP\S))\S from the lower type NP\S for the VP. It should, however, be noted that argument raising is usually not recognized as an admissible rule in standard CCG. Rules such as type raising, function composition, and argument raising are all theorems that can be derived from the more basic rules of inference in the more general, logic-based setup of TLCG including ours. One conceptual issue that remains with CCG is that it is not clear what principle determines which of the various theorems provable in TLCG are admitted as 'legal' rules in the rule set of CCG.

There is also an empirical problem (aside from the question of how to derive the distributive reading for sentences like (49) above). The treatment of scope-taking expressions in CCG, at least the one proposed in Steedman (2012), is limited in its coverage. As discussed by Barker (2007) (based on Keenan's (1992) formal proof), a compositional analysis of symmetrical predicates such as *same* and *different* requires a mechanism that goes beyond generalized quantification. In fact, as we discuss in detail in chapter 5, essentially the same problem is found in a much wider empirical domain: both 'respective' readings

[14] See also Gärtner (2012) for some discussion about quantification and scope in CCG. The treatment of distributive readings for sentences like (49) is similarly unclear in Gärtner's (2012) approach.

of conjoined and plural expressions and summative predicates such as *a total of $10,000* display properties closely resembling that of symmetrical predicates. In chapter 5, we develop a unified analysis of these phenomena building on the 'parasitic scope' analysis of symmetrical predicates proposed by Barker (2007). There is currently no explicit analysis of these phenomena in CCG, and if Keenan (1992) and Barker (2007) are right in their key analytic intuitions, this class of phenomena are likely to pose a considerable challenge to the tightly constrained, strictly surface-oriented syntax-semantics interface of CCG.

Finally, aside from the problems noted above pertaining to the 'standard' types of NCC (i.e., RNR and DCC), Gapping presents its own quite challenging problem for CCG. As we discuss in chapter 4, the analysis of Gapping in CCG by Steedman (1990) predicts only the distributive reading for auxiliaries. We refer the reader to a more detailed discussion in chapter 4, but the nature of the problem is essentially the same as above: the CCG syntax-semantics interface does not allow for a general enough treatment of scope-taking expressions, and this problem becomes particularly apparent when scope-taking expressions other than standard generalized quantifiers (such as symmetrical predicates and modal auxiliaries) are taken into account.

Given the issues noted above, we conclude that, despite the elegant analysis of the basic syntax of coordination (whose key insight we retain in its full form in our own analysis), CCG—at least as it currently stands—does not offer a general enough framework for analyzing the complex interactions between coordination and scope-taking expressions discussed in chapter 2.

3.8.2 Linear Categorial Grammar

The problem that quantification (or scope-taking more generally) poses for CCG essentially stems from the fact that the forward and backward slashes that encode directionality are not the optimal tool for characterizing the mismatch between the surface position of the quantifier and its semantic scope. In fact, as we have already noted in connection to our discussion of the vertical slash, most variants of CG, including the Lambek calculus (Lambek 1958), have essentially the same problem. The family of CGs that we call LCG here, which include the term-labelled calculus of Oehrle (1994), Abstract Categorial Grammar (de Groote 2001), Lambda Grammar (Muskens 2003), and Linear Categorial Grammar (Mihaliček and Pollard 2012), have been proposed partly in response to this empirical shortcoming of 'directional' variants of CG including the Lambek calculus.

An LCG is essentially Hybrid TLCG with only $\uparrow$ as the syntactic connective and only $\uparrow$I and $\uparrow$E as the syntactic rules. All the lexical entries involving / and \ are rewritten using $\uparrow$ and specifying word order directly in the prosodic form of the lexical entries via λ-terms as in the following lexical entry for the transitive verb *saw*:[15]

(50) $\lambda\varphi_2\lambda\varphi_1.\varphi_2 \circ \mathsf{saw} \circ \varphi_1$; **saw**; S$\uparrow$NP$\uparrow$NP

For the purpose of comparison, we present in (51) the derivation for inverse scope in LCG (compare this with (15) above).

[15]Here, for notational consistency with other parts of the book, we adopt our notation of slashes. Since $\uparrow$, the only syntactic connective in LCG, is really just linear implication, $A\uparrow B$ in our notation is more standardly written as $B \multimap A$ in the LCG literature.

(51)

$$\dfrac{\dfrac{\lambda\sigma.\sigma(\mathsf{someone});\ \mathbf{\exists}_{\mathbf{person}};\ \mathrm{S}\uparrow(\mathrm{S}\uparrow\mathrm{NP}) \qquad \dfrac{\dfrac{\dfrac{\lambda\varphi_1\lambda\varphi_2.\ \varphi_2\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1;\ \mathbf{talked\text{-}to};\ \mathrm{S}\uparrow\mathrm{NP}\uparrow\mathrm{NP} \quad \begin{bmatrix}\varphi_1;\\ x_1;\\ \mathrm{NP}\end{bmatrix}^1}{\lambda\varphi_2.\varphi_2\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1;\ \mathbf{talked\text{-}to}(x_1);\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{E} \quad \begin{bmatrix}\varphi_2;\\ x_2;\\ \mathrm{NP}\end{bmatrix}^2}{\varphi_2\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1;\ \mathbf{talked\text{-}to}(x_1)(x_2);\ \mathrm{S}}\uparrow\mathrm{E} \qquad \lambda\varphi.\varphi\circ\mathsf{yesterday};\ \mathbf{yest};\ \mathrm{S}\uparrow\mathrm{S}}{\varphi_2\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1\circ\mathsf{yesterday};\ \mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2));\ \mathrm{S}}\uparrow\mathrm{E}}{\lambda\varphi_2.\varphi_2\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1\circ\mathsf{yesterday};\ \lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2));\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{I}^2}{\mathsf{someone}\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1\circ\mathsf{yesterday};\ \mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)));\ \mathrm{S}}\uparrow\mathrm{E}}{\lambda\varphi_1.\mathsf{someone}\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1\circ\mathsf{yesterday};\ \lambda x_1.\mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)));\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{I}^1$$

$$\dfrac{\lambda\sigma.\sigma(\mathsf{everyone});\ \mathbf{\forall}_{\mathbf{person}};\ \mathrm{S}\uparrow(\mathrm{S}\uparrow\mathrm{NP}) \qquad \mathsf{someone}\circ\mathsf{talked}\circ\mathsf{to}\circ\varphi_1\circ\mathsf{yesterday};\ \mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)));\ \mathrm{S}}{\mathsf{someone}\circ\mathsf{talked}\circ\mathsf{to}\circ\mathsf{everyone}\circ\mathsf{yesterday};\ \mathbf{\forall}_{\mathbf{person}}(\lambda x_1.\mathbf{\exists}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2))));\ \mathrm{S}}\uparrow\mathrm{E}$$

Notwithstanding the elegant analysis of scope-taking phenomena available with the use of λ-binding in the prosodic component, however, LCG has its own, quite serious empirical shortcoming: unlike CCG and Lambek calculus-based variants of CG, in which there is a very simple analysis of coordination extending straightforwardly to NCC, in LCG, coordination becomes an almost intractable problem. Since this is an important empirical issue but has not received sufficient attention in the literature (but see Muskens (2001) for a cursory remark, noting but ultimately dismissing the problem), we discuss it in some detail here. A more thorough discussion addressing various partial fixes one might make in LCG, such as adding information about grammatical case (none of which generalizes properly), is found in Moot (2014) (see also Kubota (2010, Sect. 3.2.1)).

The problem can be succinctly illustrated by RNR examples such as the following:

(52) Terry hates, and Leslie likes, Robin.

Suppose we attempt to derive this example in LCG. The derivation in (53) goes through straightforwardly.

(53)
$$\dfrac{\dfrac{\lambda\varphi_1\lambda\varphi_2.\varphi_2\circ\mathsf{hates}\circ\varphi_1;\ \mathbf{hate};\ \mathrm{S}\uparrow\mathrm{NP}\uparrow\mathrm{NP} \quad [\varphi_3;x;\mathrm{NP}]^3}{\lambda\varphi_2.\varphi_2\circ\mathsf{hates}\circ\varphi_3;\ \mathbf{hate}(x);\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{E} \quad \mathsf{terry};\ \mathbf{t};\ \mathrm{NP}}{\mathsf{terry}\circ\mathsf{hates}\circ\varphi_3;\ \mathbf{hate}(x)(\mathbf{t});\ \mathrm{S}}\uparrow\mathrm{E}$$
$$\dfrac{\mathsf{terry}\circ\mathsf{hates}\circ\varphi_3;\ \mathbf{hate}(x)(\mathbf{t});\ \mathrm{S}}{\lambda\varphi_3.\mathsf{terry}\circ\mathsf{hates}\circ\varphi_3;\ \lambda x.\mathbf{hate}(x)(\mathbf{t});\ \mathrm{S}\uparrow\mathrm{NP}}\uparrow\mathrm{I}^3$$

A parallel derivation yields the category $\mathrm{S}\uparrow\mathrm{NP}$ for *Leslie likes*. But a complication arises at this point, since, unlike in CCG (or in the Lambek calculus), in LCG, the conjuncts do not correspond to strings; they are functional terms of type $\mathbf{st}\rightarrow\mathbf{st}$. Thus, they cannot be directly concatenated to form the coordinated string. One might think that assigning the following type of lexical entry for the conjunction word *and* (of type $(\mathbf{st}\rightarrow\mathbf{st})\rightarrow(\mathbf{st}\rightarrow\mathbf{st})\rightarrow(\mathbf{st}\rightarrow\mathbf{st})$) would work:

(54) $\lambda\sigma_1\lambda\sigma_2\lambda\varphi.\sigma_2(\boldsymbol{\epsilon})\circ\mathsf{and}\circ\sigma_1(\varphi);\ \sqcap;\ (\mathrm{S}\uparrow\mathrm{NP})\uparrow(\mathrm{S}\uparrow\mathrm{NP})\uparrow(\mathrm{S}\uparrow\mathrm{NP})$

Applying this functor to the two conjuncts indeed yields the following sign for the whole coordinate structure:

(55) $\lambda\varphi.\mathsf{terry} \circ \mathsf{hates} \circ \boldsymbol{\epsilon} \circ \mathsf{and} \circ \mathsf{leslie} \circ \mathsf{likes} \circ \varphi;\ \lambda x.\mathbf{hate}(x)(\mathbf{t}) \wedge \mathbf{like}(x)(\mathbf{l});\ \mathrm{S}\uparrow\mathrm{NP}$

to which the object NP *Robin* can be given as an argument to complete the derivation.

This analysis, however, overgenerates in a quite serious way. To see this, note that the following can also be derived as a well-formed sign with category S↑NP in LCG via hypothetical reasoning:

(56) $\lambda\varphi_1.\varphi_1 \circ \mathsf{likes} \circ \mathsf{robin};\ \mathbf{like}(\mathbf{r})(x);\ \mathrm{S}\uparrow\mathrm{NP}$

Conjoining this with the same first conjunct *Terry hates* above yields the following expression:

(57) $\lambda\varphi.\mathsf{terry} \circ \mathsf{hates} \circ \boldsymbol{\epsilon} \circ \mathsf{and} \circ \varphi \circ \mathsf{likes} \circ \mathsf{robin};\ \lambda x.\mathbf{hate}(x)(\mathbf{t}) \wedge \mathbf{like}(\mathbf{r})(x);\ \mathrm{S}\uparrow\mathrm{NP}$

By giving *Leslie* as an argument to this functor, we have an analysis for the sentence *Terry hates and Leslie likes Robin* to which the meaning 'Terry hates Leslie and Leslie likes Robin' is assigned, which is obviously wrong. The problem, of course, is that what in a directional CG would be two distinct types S/NP and NP\S are conflated as both instances of S↑NP in LCG, and therefore there is no apparent way to avoid conjoining *Terry hates* with *likes Robin*.[16]

A problem of essentially the same nature arises with even more basic cases of constituent coordination such as the following, with an even more embarrassing effect:

(58) John caught and ate the fish.

Note that, in LCG, the following two signs are interderivable (hypothesizing an object and subject NPs and withdrawing them in the opposite order derives (59b) from (59a)):

(59) a. $\lambda\varphi_1\lambda\varphi_2.\varphi_2 \circ \mathsf{ate} \circ \varphi_1;\ \mathbf{ate};\ \mathrm{S}\uparrow\mathrm{NP}\uparrow\mathrm{NP}$

b. $\lambda\varphi_2\lambda\varphi_1.\varphi_2 \circ \mathsf{ate} \circ \varphi_1;\ \lambda x\lambda y.\mathbf{ate}(y)(x);\ \mathrm{S}\uparrow\mathrm{NP}\uparrow\mathrm{NP}$

Then, a conjunction entry in (60) for coordination of transitive verbs of type S↑NP↑NP can license (58), at the expense of predicting that the sentence is ambiguous in all of the four readings in (61).

(60) $\lambda\sigma_1\lambda\sigma_2\lambda\varphi_1\lambda\varphi_2.\varphi_2 \circ \sigma_2(\boldsymbol{\epsilon})(\boldsymbol{\epsilon}) \circ \mathsf{and} \circ \sigma_1(\boldsymbol{\epsilon})(\boldsymbol{\epsilon}) \circ \varphi_1;\ \sqcap;\ (\mathrm{S}\uparrow\mathrm{NP}\uparrow\mathrm{NP})\uparrow(\mathrm{S}\uparrow\mathrm{NP}\uparrow\mathrm{NP})\uparrow(\mathrm{S}\uparrow\mathrm{NP}\uparrow\mathrm{NP})$

(61) a. $\mathbf{caught}(\mathbf{j})(\mathbf{the\text{-}fish}) \wedge \mathbf{ate}(\mathbf{j})(\mathbf{the\text{-}fish})$

b. $\mathbf{caught}(\mathbf{j})(\mathbf{the\text{-}fish}) \wedge \mathbf{ate}(\mathbf{the\text{-}fish})(\mathbf{j})$

c. $\mathbf{caught}(\mathbf{the\text{-}fish})(\mathbf{j}) \wedge \mathbf{ate}(\mathbf{j})(\mathbf{the\text{-}fish})$

d. $\mathbf{caught}(\mathbf{the\text{-}fish})(\mathbf{j}) \wedge \mathbf{ate}(\mathbf{the\text{-}fish})(\mathbf{j})$

[16] Note that, even though Hybrid TLCG incorporates LCG as its subsystem, the type of overgeneration in LCG in coordination discussed here does not arise in Hybrid TLCG. The problem with LCG in a nutshell is that since the system allows for only the vertical slash ↑, which does not encode linear order, one is forced to specify the conjunction category for functional expressions (such as verbs) in terms of ↑, losing control over linear order. In Hybrid TLCG (as in CCG), the conjunction category is specified in terms of directional slashes (except for the case of Gapping, which receives a different treatment due to the fact that the 'gap' is medial), and conjunction entries such as (54) and (60) discussed here are not posited.

3.9 Computational issues

Although Hybrid TLCG is primarily meant to be a theory of competence grammar, and, moreover, we don't take practical applicability in parsing to be one of our primary goals (this sets it apart from CCG), considerations of computational properties is an important issue for formally explicit frameworks of grammar, including, most notably, CG. Moreover, the existence of an actually implemented parser—even one that does not compete realistically with efficient parsers that are designed to parse real text—would be useful in the context of grammar checking, and can potentially provide a quite valuable tool for working linguists. In this section, we address these computational issues briefly.

It is known that a certain restricted form of Hybrid TLCG—specifically, one which eliminates empty operators and polymorphic specifications of lexical entries—is decidable and NP complete (Moot 2014), just like related approaches in TLCG such as Displacement Calculus (Morrill et al. 2011) and NL_λ (Barker and Shan 2015). Decidability is an important property when considering the formal computational properties of a grammatical framework, since it ensures that the search space is limited for any given string.

As will become clear in the following chapters, we formulate analyses of linguistic phenomena making somewhat extensive use of empty operators and polymorphic specifications. Whether these empty operators and polymorphic specifications of lexical entries can be eliminated without sacrificing the generality of the linguistic analyses proposed below too much is currently an open question. There are some points worth noting in this connection. First, empty operators introduced for the analyses of certain syntactic phenomena are relatively 'harmless' in that they can be lexicalized in the meanings of words that have actual prosodic content (which reduces the problem to lexical redundancy; at least from a computational perspective, this would be by far a much less costly problem than losing the decidability of the entire grammar). This is the case with the analysis of pseudogapping we propose in chapter 6, where the relevant operator can be lexicalized in the meaning of the auxiliary verb.[17] However, there are some instances of empty operators which do not seem to lend themselves to lexicalization as easily. An example of this is the empty operators that we extensively rely on in our analysis of 'respective' readings and related phenomena in chapter 5.[18] In general, there does seem to be a trade-off between computational concerns and elegance of linguistic analysis in the use of empty operators. In the literature of plurality in theoretical linguistics in which computational issues are not the foremost concerns, empty operators are already extensively employed even for phenomena that are (at least in certain respects) less complex than those that we deal with in chapter 5. Clearly, something more needs to be said on this important issue, given that human speakers have no trouble understanding sentences containing plurals, *respectively*, and symmetrical and summative predicates (at least simple ones). Unfortunately, addressing this question is beyond the scope of the current work and we have to leave it for future study.

Regarding the question of a working parser, there is a notable progress in this domain. Moot (2014) proves decidability and NP completeness of Hybrid TLCG through a translation of Hybrid TLCG into first-order linear logic. As a by-product of this work, a parser of

[17] This chapter is not yet written. See Kubota and Levine (2015b) for the analysis of pseudogapping.

[18] This chapter is not yet written. See Kubota and Levine (2014c) for the analysis of 'respective' readings and related phenomena.

Hybrid TLCG written in SWI Prolog, called LinearOne, is made available by Richard Moot. The repository (`https://github.com/RichardMoot/LinearOne`) currently contains a toy grammar that can parse some sample sentences (mostly from the two articles on Gapping—Kubota and Levine (2012) and Kubota and Levine (2013b)) with a brief documentation. The parser runs on any platform that has an implementation of SWI Prolog and a relatively recent version of LaTeX to graphically show the parse outputs in pdf. Lexical entries can be added to the grammar easily, and by doing so, one can, for example, check the linguistic analyses we propose in the current monograph by oneself (but due to the issues that they cause on decidability, polymorphic specifications need to be instantiated and empty operators need to be removed by lexicalization). We believe that the LinearOne parser, with the wide empirical coverage that Hybrid TLCG offers, can supply the much needed tool for grammar checking not (only) for computational linguists, but primarily for working syntacticians and semanticists. In this respect, the similarity between Hybrid TLCG and the mainstream derivational architecture of grammar is especially noteworthy. Given the transparent correspondence between the two (in most cases), an analysis formulated assuming the standard derivational architecture can be straightforwardly translated into Hybrid TLCG, and then one can check the predictions of one's analysis by implementing it in the LinearOne parser. In most practical cases, this involves just adding a handful of lexical entries for the key lexical items.

3.10 Appendix: Formal fragment of Hybrid TLCG

3.10.1 Syntactic, semantic and prosodic types

Syntactic types in Hybrid TLCG are defined as follows:

(62) Atomic types include (at least) NP, N and S.

(63) Directional types
   a. An atomic type is a directional type.
   b. If A and B are directional types, then (A/B) is a directional type.
   c. If A and B are directional types, then $(B\backslash A)$ is a directional type.
   d. Nothing else is a directional type.

(64) Syntactic types
   a. A directional type is a syntactic type.
   b. If A and B are syntactic types, then $(A{\uparrow}B)$ is a syntactic type.
   c. Nothing else is a syntactic type.

We omit outermost parentheses and parentheses for a sequence of the same type of slash, assuming that / and $\uparrow$ are left associative, and \ right associative. Thus, S/NP/NP, NP\NP\S and S$\uparrow$NP$\uparrow$NP are abbreviations of ((S/NP)/NP), (NP\(NP\S)) and ((S$\uparrow$NP)$\uparrow$NP), respectively.

Note that the algebra of syntactic types is *not* a free algebra generated over the set of atomic types with the three binary connectives /, \, and $\uparrow$. Specifically, given the definitions in (62)–(64), in Hybrid TLCG, a vertical slash cannot occur 'under' a directional

slash. Thus, S/(S↑NP) is not a well-formed syntactic type. One way to make sense of this restriction is to think of it as a 'filter' on uninterpretable prosodic objects. An expression with syntactic type X/(Y↑Z) would have to concatenate a string to the left of a functor of type $\mathbf{st} \to \mathbf{st}$, but it does not make sense (at least if one takes the 'meanings' of slashes literally) to 'concatenate' a string to the left of a *function* from strings to strings since concatenation is an operation defined only on strings.

The functions Sem and Pros, which map syntactic types to semantic and prosodic types, can be defined as follows:

(65) a. For atomic syntactic types,
$\mathsf{Sem}(\mathrm{NP}) = e$, $\mathsf{Sem}(\mathrm{S}) = t$, $\mathsf{Sem}(\mathrm{N}) = e \to t$

b. For complex syntactic types,
$\mathsf{Sem}(A/B) = \mathsf{Sem}(B\backslash A) = \mathsf{Sem}(A\uparrow B) = \mathsf{Sem}(B) \to \mathsf{Sem}(A)$.

(66) a. For any directional type A, $\mathsf{Pros}(A) = \mathbf{st}$ (with $\mathbf{st}$ for 'strings').

b. For any complex syntactic type $A\uparrow B$ involving the vertical slash ↑,
$\mathsf{Pros}(A\uparrow B) = \mathsf{Pros}(B) \to \mathsf{Pros}(A)$.

Note in particular that, corresponding to the asymmetry between /, \, and ↑ in the definition of syntactic types, the two types of slashes behave differently in the mapping from syntactic to prosodic types. Specifically, for the mapping from syntactic types to prosodic types, only the vertical slash ↑ is effectively interpreted as functional. Thus, for example, $\mathsf{Sem}(\mathrm{S}\uparrow(\mathrm{S/NP})) = (e \to t) \to t$ whereas $\mathsf{Pros}(\mathrm{S}\uparrow(\mathrm{S/NP})) = \mathbf{st} \to \mathbf{st}$.

The prosodic calculus is a λ-calculus with constants of type $\mathbf{st}$ (which includes ordinary string prosodies such as john, walks, etc., and the empty string $\boldsymbol{\epsilon}$) and a binary connective $\circ$ for string concatenation. In addition to beta equivalence, the following axioms hold in the prosodic calculus:

(67) a. $\boldsymbol{\epsilon} \circ a \equiv a$ (left identity)

b. $a \circ \boldsymbol{\epsilon} \equiv a$ (right identity)

(68) $a \circ (b \circ c) \equiv (a \circ b) \circ c$ (associativity)

In the derivations, we omit parentheses for $\circ$ and implicitly assume associativity.

3.10.2 Sample lexicon

(69) a. john; $\mathbf{j}$; NP

b. mary; $\mathbf{m}$; NP

c. walks; $\mathbf{walk}$; NP\S

d. loves; $\mathbf{love}$; (NP\S)/NP

e. $\lambda\sigma.\sigma(\mathsf{everyone})$; $\forall_{\mathbf{person}}$; S↑(S↑NP)

f. $\lambda\varphi\lambda\sigma.\sigma(\mathsf{every} \circ \varphi)$; $\forall$; S↑(S↑NP)↑N

g. $\lambda\sigma.\sigma(\mathsf{must})$; $\lambda\mathscr{F}.\Box\mathscr{F}(\mathsf{id}_{et})$; S↑(S↑(VP/VP))
(where $\mathsf{id}_{et} = \lambda P_{et}.P$)

For strings in prosodic representations we use $\mathsf{sans\text{-}serif}$; semantic constants are written in **bold face**; for prosodic variables we use Greek letters $\varphi_1, \varphi_2, \ldots$ (type $\mathbf{st}$ for string); $\sigma_1, \sigma_2, \ldots$ (type $\mathbf{st} \to \mathbf{st}$, $\mathbf{st} \to \mathbf{st} \to \mathbf{st}$, etc.); $\tau_1, \tau_2, \ldots$ (type $(\mathbf{st} \to \mathbf{st}) \to \mathbf{st}$, etc.), and for semantic variables roman italics ($x, y, z, p, q, \ldots, P, Q, R, \ldots$); calligraphic letters ($\mathcal{U}, \mathcal{V}, \mathcal{W}, \ldots$) are

invariably used for variables with polymorphic types and copperplate letters ($\mathscr{P}, \mathscr{Q}, \ldots$) for higher-order variables of fixed types.

3.10.3 Syntactic rules

Logical rules

(70)

Connective	Introduction	Elimination
/	$\dfrac{\begin{array}{c}\vdots\ \vdots\ [\varphi;\ x;\ A]^n\ \vdots\ \vdots \\ \vdots\ \vdots\qquad\vdots\ \vdots\qquad\vdots\ \vdots\end{array}}{b \circ \varphi;\ \boldsymbol{\mathcal{F}};\ B}\,/\mathrm{I}^n$ $b;\ \lambda x.\boldsymbol{\mathcal{F}};\ B/A$	$\dfrac{a;\ \boldsymbol{\mathcal{F}};\ A/B \quad b;\ \boldsymbol{\mathcal{G}};\ B}{a \circ b;\ \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\ A}\,/\mathrm{E}$
\	$\dfrac{\begin{array}{c}\vdots\ \vdots\ [\varphi;\ x;\ A]^n\ \vdots\ \vdots \\ \vdots\ \vdots\qquad\vdots\ \vdots\qquad\vdots\ \vdots\end{array}}{\varphi \circ b;\ \boldsymbol{\mathcal{F}};\ B}\,\backslash\mathrm{I}^n$ $b;\ \lambda x.\boldsymbol{\mathcal{F}};\ A\backslash B$	$\dfrac{b;\ \boldsymbol{\mathcal{G}};\ B \quad a;\ \boldsymbol{\mathcal{F}};\ B\backslash A}{b \circ a;\ \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\ A}\,\backslash\mathrm{E}$
↑	$\dfrac{\begin{array}{c}\vdots\ \vdots\ [\varphi;\ x;\ A]^n\ \vdots\ \vdots \\ \vdots\ \vdots\qquad\vdots\ \vdots\qquad\vdots\ \vdots\end{array}}{b;\ \boldsymbol{\mathcal{F}};\ B}$ $\lambda\varphi.b;\ \lambda x.\boldsymbol{\mathcal{F}};\ B\uparrow A\ \uparrow\mathrm{I}^n$	$\dfrac{a;\ \boldsymbol{\mathcal{F}};\ A\uparrow B \quad b;\ \boldsymbol{\mathcal{G}};\ B}{a(b);\ \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\ A}\,\uparrow\mathrm{E}$

All of the three slashes are linear. That is, the three connectives can bind only one occurrence of a hypothesis at a time. Note also that the prosodic labels in Hybrid TLCG are not proof terms, but rather are used for the purpose of narrowing down the set of possible derivations. Specifically, the applicability of the Introduction rule for / (\) is conditioned by the presence of the variable φ at the right (left) periphery in the prosody of the premise.

Non-logical rule

(71) P-interface rule
$\dfrac{\varphi_0;\ \boldsymbol{\mathcal{F}};\ A}{\varphi_1;\ \boldsymbol{\mathcal{F}};\ A}\,\mathrm{PI}$
——where φ_0 and φ_1 are equivalent terms in the prosodic calculus

The P-interface rule is the analog of the structural rules in the more standard variants of TLCG (Morrill 1994; Moortgat 1997). This rule is used for applying beta-reduction to prosodic terms obtained by function application via the ↑E rule. For example, a more 'pedantic' and technically precise (but cumbersome) version of the quantifier scope derivation in (15) would contain the following proof steps (note the step-by-step β-reduction in the prosodic component mediated by PI):

(72)
$$\dfrac{\dfrac{\dfrac{\vdots \quad \vdots}{\begin{array}{l}\lambda\sigma.\sigma(\textsf{someone}); \quad \lambda\varphi_2.\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{yesterday};\\ \mathfrak{A}_{\mathbf{person}}; \mathrm{S}\uparrow(\mathrm{S}\uparrow\mathrm{NP}) \quad \lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2)); \mathrm{S}\uparrow\mathrm{NP}\end{array}}\uparrow\mathrm{E}}{\begin{array}{l}\lambda\sigma.[\sigma(\textsf{someone})](\lambda\varphi_2.\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{yesterday});\\ \mathfrak{A}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2))); \mathrm{S}\end{array}}\mathrm{PI}}{\begin{array}{l}\lambda\varphi_2.[\varphi_2 \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{yesterday}](\textsf{someone});\\ \mathfrak{A}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2))); \mathrm{S}\end{array}}\mathrm{PI}$$
$$\textsf{someone} \circ \textsf{talked} \circ \textsf{to} \circ \varphi_1 \circ \textsf{yesterday};\ \mathfrak{A}_{\mathbf{person}}(\lambda x_2.\mathbf{yest}(\mathbf{talked\text{-}to}(x_1)(x_2))); \mathrm{S}$$
$$\vdots \quad \vdots$$

In practice, we omit explicitly writing the application of this rule to avoid cluttering the derivations.

One way to understand the (apparent) asymmetry between the $\uparrow$I rule and the /I and \I rules as formulated above as to the explicit presence of λ-binding in the prosodic component in the former but not in the latter two is to think of the /I and \I rules as abbreviations of the following proof steps:

(73)
$$\dfrac{\dfrac{\dfrac{\vdots\ \vdots\ \underline{[\varphi;\ x;\ A]^n}\ \vdots\ \vdots}{\vdots\ \vdots \qquad \vdots\ \vdots \qquad \vdots\ \vdots}}{b \circ \varphi;\ \boldsymbol{\mathcal{F}};\ B}/\mathrm{I}^n}{\lambda_{\mathsf{r}}\varphi.[b \circ \varphi](\boldsymbol{\epsilon});\ \lambda x.\boldsymbol{\mathcal{F}};\ B/A}\mathrm{PI}}{b;\ \lambda x.\boldsymbol{\mathcal{F}};\ B/A}\mathrm{PI}$$

That is, there is actually lambda binding (by 'left' and 'right' lambdas) in the prosody in the /I and \I rules, but the functional lambda terms obtained are immediately applied to the empty string to 'close off' the gap. On this view, the /I and \I rules are not so different from the $\uparrow$I rule after all, but the key difference is that, unlike the $\uparrow$I rule, the /I and \I rules are immediately followed by an obligatory application to the empty string, so that a string prosody rather than a functional prosody is obtained.

3.10.4 Relationship between Hybrid TLCG and other variants of CG

The relationship between Hybrid TLCG and other variants of CG discussed in this chapter can be roughly summarized as follows:[19]

(74)

	Rules
AB	/E, \E
Lambek calculus	/E, \E, /I, \I
LCG	$\uparrow$E, $\uparrow$I
NL_λ	/E, \E, $\uparrow$E, $\uparrow$I
Hybrid TLCG	/E, \E, $\uparrow$E, /I, \I, $\uparrow$I

[19]Barker and Shan's (2015) NL_λ is less general than this subset of Hybrid TLCG in that it doesn't admit (an analog of) higher-order prosodic variables. Also, NL_λ actually employs the continuation slashes $\mathbin{/\!\!/}$ and $\mathbin{\backslash\!\!\backslash}$ in place of $\uparrow$ whose technical implementation is quite different from $\uparrow$ in our system, so it should be noted that the comparison here is only a rough comparison at a conceptual level.

Bibliography

Ajdukiewicz, Kazimierz. 1935. Die syntaktische Konnexität. In S. McCall, ed., *Polish Logic 1920–1939*, 207–231. Oxford: OUP. Translated from *Studia Philosophica*, 1, 1–27.

Baldridge, Jason. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

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

Bar-Hillel, Yehoshua, Chaim Gaifman, and Eliyahu Shamir. 1960. On categorial and phrase structure grammars. *Bulletin of the Research Council of Israel* 9F:1–16. Also as Chapter 8 in Bar-Hillel (1964) *Language and Information*, Addison Weseley.

Barker, Chris. 2007. Parasitic scope. *Linguistics and Philosophy* 30:407–444.

Barker, Chris and Chung-chieh Shan. 2015. *Continuations and Natural Language*. Oxford: OUP.

Bernardi, Raffaella. 2002. *Reasoning with Polarity in Categorial Type Logic*. Ph.D. thesis, University of Utrecht.

Borsley, Robert D. and Kersti Börjars, eds. 2011. *Non-Transformational Syntax*. Wiley-Blackwell.

Cooper, Robin. 1983. *Quantification and Syntactic Theory*, vol. 21 of *Synthese Language Library - Text and Studies in Linguistics and Philosophy*. Dordrecht, Boston and London: D. Reidel Publishing Company.

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. 1988. Type raising, functional composition, and non-constituent conjunction. In R. T. Oehrle, E. Bach, and D. Wheeler, eds., *Categorial Grammars and Natural Language Structures*, 153–197. Dordrecht: D. Reidel Publishing Company.

Dowty, David R. 1996. Toward a minimalist theory of syntactic structure. In H. Bunt and A. van Horck, eds., *Discontinuous Constituency*, vol. 6 of *Natural Language Processing*, 11–62. Berlin, New York: Mouton de Gruyter.

Gärtner, Hans-Martin. 2012. Function composition and the linear local modeling of extended negscope. In A. Alexiadou, T. Kiss, and G. Müller, eds., *Local Modelling of Non-Local Dependencies in Syntax*, 337–352. Berlin: de Gruyter.

Hendriks, Herman. 1993. *Studied Flexibility*. Ph.D. thesis, University of Amsterdam, Amsterdam.

Hepple, Mark. 1990. *The Grammar and Processing of Order and Dependency: A Categorial Approach*. Ph.D. thesis, University of Edinburgh.

Keenan, Edward L. 1992. Beyond the Frege boundary. *Linguistics and Philosophy* 15(2):199–221.

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

Kubota, Yusuke. 2014. The logic of complex predicates: A deductive synthesis of 'argument sharing' and 'verb raising'. *Natural Language and Linguistic Theory* 32(4):1145–1204.

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 and Robert Levine. 2012. Gapping as like-category coordination. In D. Béchet and A. Dikovsky, eds., *Logical Aspects of Computational Linguistics 2012*, 135–150. Heidelberg: Springer.

Kubota, Yusuke and Robert Levine. 2013a. Coordination in Hybrid Type-Logical Categorial Grammar. In *OSU Working Papers in Linguistics*, vol. 60, 21–50. Department of Linguistics, Ohio State University.

Kubota, Yusuke and Robert Levine. 2013b. Determiner gapping as higher-order discontinuous constituency. In G. Morrill and M.-J. Nederhof, eds., *Proceedings of Formal Grammar 2012 and 2013*, 225–241. Heidelberg: Springer.

Kubota, Yusuke and Robert Levine. 2014a. Gapping as hypothetical reasoning. To appear in *NLLT*, available at `http://ling.auf.net/lingbuzz/002123`.

Kubota, Yusuke and Robert Levine. 2014b. Pseudogapping as pseudo-VP ellipsis. In N. Asher and S. Soloviev, eds., *Logical Aspects of Computational Linguistics 2014*, 122–137. Heidelberg: Springer.

Kubota, Yusuke and Robert Levine. 2014c. The syntax-semantics interface of 'respective' predication: A unified analysis in Hybrid Type-Logical Categorial Grammar. To appear in *NLLT*, MS., University of Tsukuba and Ohio State University, available at `http://ling.auf.net/lingbuzz/002150`.

Kubota, Yusuke and Robert Levine. 2015a. Against ellipsis: Arguments for the direct licensing of 'non-canonical' coordinations. To appear in *Linguistics and Philosophy*, MS., University of Tsukuba and Ohio State University, available at `http://ling.auf.net/lingbuzz/002214`.

Kubota, Yusuke and Robert Levine. 2015b. Pseudogapping as pseudo-VP ellipsis. MS., University of Tsukuba and Ohio State University, available at `http://ling.auf.net/lingbuzz/002504`.

Lambek, Joachim. 1958. The mathematics of sentence structure. *American Mathematical Monthly* 65:154–170.

Martin, Scott. 2013. *The Dynamics of Sense and Implicature*. Ph.D. thesis, Ohio State University.

Martin, Scott and Carl Pollard. 2014. A dynamic categorial grammar. In G. Morrill, R. Muskens, R. Osswald, and F. Richter, eds., *Proceedings of Formal Grammar 2014*, 138–154. Heidelberg: Springer.

Mihaliček, Vedrana and Carl Pollard. 2012. Distinguishing phenogrammar from tectogrammar simplifies the analysis of interrogatives. In P. de Groote and M.-J. Nederhof, eds., *Formal Grammar 2010/2011*, 130–145. Heidelberg: Springer.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, J. M. Moravcsik, and P. Suppes, eds., *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, 221–242. Dordrecht, Holland: D. Reidel.

Moortgat, Michael. 1997. Categorial Type Logics. In J. van Benthem and A. ter Meulen, eds., *Handbook of Logic and Language*, 93–177. Amsterdam: Elsevier.

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

Moot, Richard. 2014. Hybrid type-logical grammars, first-order linear logic and the descriptive inadequacy of Lambda grammars. MS., Laboratoire Bordelais de Recherche en Informatique.

Morrill, Glyn. 1994. *Type Logical Grammar: Categorial Logic of Signs*. Dordrecht: Kluwer.

Morrill, Glyn and 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, and Mario Fadda. 2011. The displacement calculus. *Journal of Logic, Language and Information* 20:1–48.

Muskens, Reinhard. 2001. Categorial Grammar and Lexical-Functional Grammar. In M. Butt and

T. H. King, eds., *The Proceedings of the LFG '01 Conference*. University of Hong Kong.

Muskens, Reinhard. 2003. Language, lambdas, and logic. In G.-J. Kruijff and R. Oehrle, eds., *Resource Sensitivity in Binding and Anaphora*, 23–54. Dordrecht: Kluwer.

Muskens, Reinhard. 2007. Separating syntax and combinatorics in categorial grammar. *Research on Language and Computation* 5(3):267–285.

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

Partee, Barbara and Mats Rooth. 1983. Generalized conjunction and type ambiguity. In R. Bäuerle, C. Schwarze, and A. von Stechow, eds., *Meaning, Use, and Interpretation of Language*, 361–383. Berlin: Walter de Gruyter.

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 and E. Allyn Smith. 2012. A unified analysis of *the same*, phrasal comparatives and superlatives. In *Proceedings of SALT 2012*, 307–325.

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

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

Steedman, Mark. 1990. Gapping as constituent coordination. *Linguistics and Philosophy* 13(2):207–263.

Steedman, Mark. 1996. *Surface Structure and Interpretation*. Cambridge, Mass.: MIT Press.

Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, Mass.: MIT Press.

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

Vermaat, Willemijn. 2005. *The Logic of Variation. A Cross-Linguistic Account of Wh-Question Formation*. Ph.D. thesis, Utrecht University.

Worth, Chris. 2014. The phenogrammar of coordination. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, 28–36. Gothenburg, Sweden: Association for Computational Linguistics.